POWERFUL LIBRARY

HANDY PERSONAL COMPUTER
CASIO FP-200

CETL

BASIC

CASIO

PREFACE

The recent popularity of the portable computer is remarkable. The implication is that portability, providing extra versatility to the personal computer, has made a strong appeal to computer users who place emphasis on ease of operation. The portable computer has a typewriter-style keyboard like the one used with an ordinary desk-top computer. It also has a display panel of moderate size.

The FP-200, on which the programs presented in this booklet run, is a portable computer featuring low costant high performance. The advent of the FP-200 with its innovative new features has proved that portable computers have great potential for further growth.

The greatest advantage of the FP-200 is that a simple table language, called CETL, is supplied as a standard feature. CETL makes tabulation work extremely easy. It permits summing up items in horizontal rows and in vertical columns or storing data in a table by a single command.

Of course, the same tabulation operations can be performed by a BASIC program, but substantial time is requried to write the program. In this light, the fact that CETL requires no programming is of great significance.

With the FP-200, any table prepared by CETL can be easily called by a program written in BASIC. This feature is called the link between CETL and BASIC. The CETL-BASIC link is really an important concept.

The reason is this: the link permits the operator to use a table prepared by CETL in interactive mode using a BASIC program, or to compare or adjoin data input by a BASIC program with data prepared by CETL, etc.

CETL permits flexible table construction by using 16 types of simple commands. The commands include the A command to determine the direction of data input, the I, D, and M commands to add, delete, and move items, the S command to rearrange data, the F command to perform conditional retrieval, and the C command to perform calculations. These commands permit performing the desired operations on specified "items" or "records".

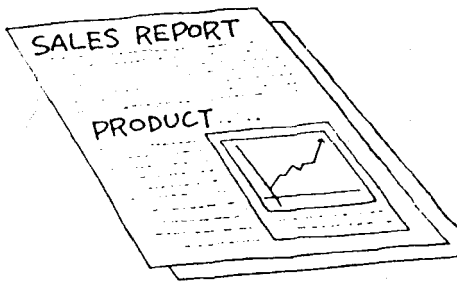
"Records" are sets of entries in horizontal rows, and "Items" are sets of entries

Business Programs

REGRESSION ANALYSIS
(SALES FORECASTING)

Program Outline

Regression analysis can often be used to make a forecast of a future trend based on relevant data. Using regression analysis, this program forecasts this year's sales based on last year's sales. In this program, the forecast is made in weekly units.



Directions

1. Reading data from CETL

The data table used by this program can be read from CETL. Create the data table in CETL as follows:

(1) Input this year's weekly sales in item 1 (records 1 to 52) and last year's weekly sales in item 2 (records 1 to 52).

(2) Reserve records 53 and 54 of items 1 and 2 for the values X, X2, Y#, and XY# used for the link between CETL and BASIC.

2. Executing the program

When the program is started, the prompt "DATA INPUT CETL(1)/BASIC(2)" is displayed, asking you whether the last year's data is to be input from the data table previously stored in CETL or by a BASIC program. To input the data from CETL, enter 1. To input the data using a BASIC program, enter 2.

If you entered 1, the messages "BEING COMPUTED" and "WHICH WEEK" are displayed. Now, enter the week (n-th week) for which the sales forecast is to be made. The result of the forecast is then output. After that, the prompt "WHICH WEEK?" is displayed again. You can continue the sales forecasting by specifying an appropriate week. At the end of the forecasting for the final week, enter 0 and the program will terminate.

If you entered 2, you are requested to input last year's sales data starting from the first week. After you input all the data, enter E, and the prompts "BEING COMPUTED" and "WHICH WEEK" are displayed. Now, enter the week for which the sales forecast is to be made. The result of the forecast is output in the same manner as when you enter 1.

LIST

684 Bytes

```
   10 CLEAR:CLS:PRINT"DATA INPUT                CETL(1)/B
ASIC(2)"
   20 INPUT D$
   30 IF D$="1" THEN GOSUB 100
   40 IF D$="2" THEN GOSUB 200
   50 K#=(12*XY#-X*Y#)/(12*X2-X^2)
   60 H#=(X2*Y#-X*XY#)/(12*X2-X^2)
   65 CLS
   70 INPUT "Which week ";J:IF J=0 THEN END
   80 V#=K#*J+H#
   90 PRINTV#:PRINT USING"          ##th week";J:PRI
NT:GOTO 70
  100 PRINT"Being computed"
  110 X=SUMRC(1,52):Y#=SUMRC(1,52)
  120 FOR I=1 TO 52
  130 X2=RC(I)^2+X2
  140 XY#=I*RC(I)+XY#
  150 NEXT I
  160 RETURN
  200 OPTION BASE 1:DIM X(80),Y#(80)
  210 I=I+1
  220 PRINT USING"##th week                    of last y
ear";I
  230 INPUT "SALES (END'E')";A$
  240 IF A$<>"E" THEN Y#(I)=VAL(A$):X(I)=I:GOTO 210
  250 T=I-1
  260 PRINT"Being computed"
  270 FOR I=1 TO T
  280 X=X(I)+X:X2=X(I)^2+X2:Y#=Y#(I)+Y#:XY#=X(I)*Y#
(I)+XY#
  290 NEXT I
  300 RETURN
```

DEBUG

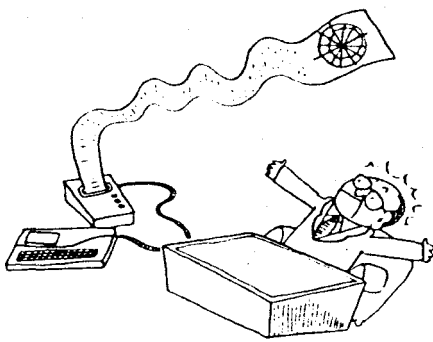
This program can be used for various types of forecasting. However, widely fluctuating figures will not follow a regression line. In this case, the figures forecast by this program may not represent the actual figures with fair accuracy.

BUSINESS ANALYSIS AND CHARTING

Program Outline

The performance of a business enterprise is accurately reflected by its financial statements (balance sheet, income statement, etc.). The FP-200 can be conveniently used by the banker, financial auditor, business consultant, etc., because of its capability to output the performance of a business enterprise in the form of graphics. For this purpose, it is only necessary to input major indicators appearing on the financial statements.

This program permits the plotter to output a chart, data tables, diagnostic data, etc., when 13 major indicators used for business analysis are input.



Data Input

(1) Preparing the data

Two methods of data input are available: one is to input data from the appropriat data table prepared in CETL, and the other is to input data in response to requests fo input made by a BASIC program.

Either method can be used to produce the same result. Since the chart is for comparin a particular set of figures, the data to be prepared is determined by the purpose of th comparison — comparing the performance figures for the current term with those fo another term of a particular firm, comparing two (or more) companies with respect t particular performance figures, and so on.

Up to five charts can be plotted on the same graph for comparison. When only dat for a single chart is input, the program merely plots a regular 13-sided polygon becaus no comparison can be made.

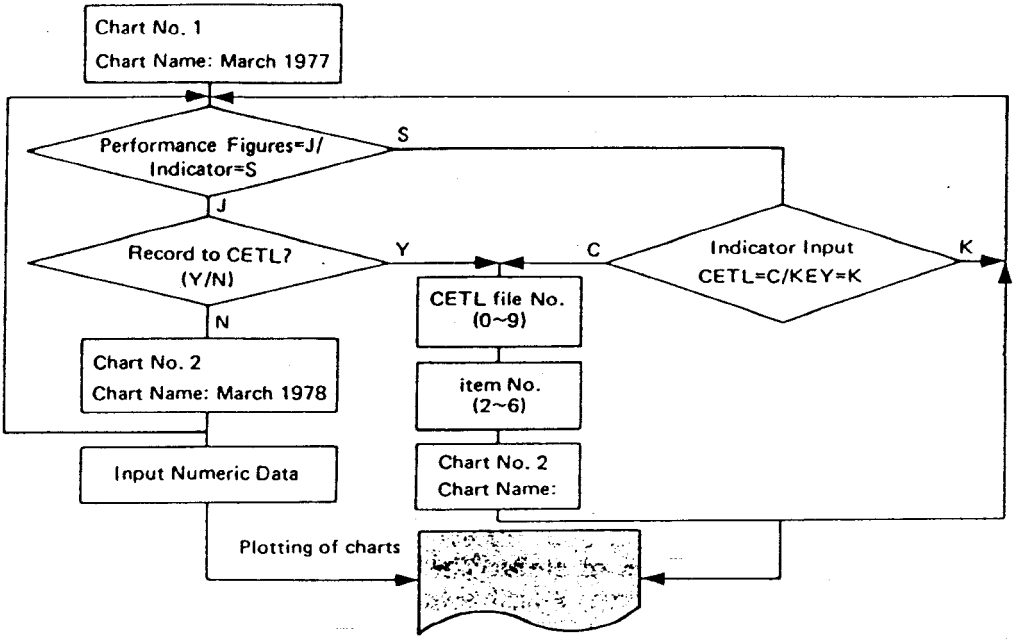
(2) Starting the program

When the program is started, the message "INDICATORS CAN BE READ FROM OF WRITTEN TO CETL" is displayed. Press any one key, and the message "UP TO FIVE CHARTS CAN BE PLOTTED. SPECIFY NUMBER OF CHARTS, CHART NAMES AND INPUT METHOD. NUMBER OF CHARTS —" is displayed. Now, specify th number of charts to be plotted, the chart names (e.g., fiscal periods or company name used for comparison), and the method of input (when the result of calculation on each data is already known, indicator values, say, percentage figures, can be input; otherwise actual performance figures should be input).

(3) When input of performance figures is selected

After the chart names (company names, fiscal periods, etc.) are input, the promp "PERFORMANCE FIGURES=J/INDICATOR=S _" is displayed. To input perfor

low chart for data input



ance figures, enter J. To input indicator values, enter S. Enter J when testing this ogram fro the first time.

hen J is entered, the prompt "STORE TO CETL? (Y/N) _" is displayed. This is to bulate the input data using CETL and store it in the CETL area. If you wish to select , you should check to see that space is available in the CETL area. After entering Y, put the file area number and item number as shown in the flow chart for data input. you entered N (instead of Y), the prompt "CHART NO.2 CHART NAME:" is splayed. This prompt is repeated the same number of times are the number of charts. fter you input all the chart names, you are requested to input the relevant data urrent assets, inventories, . . . , employees) for each chart name.

put the appropriate data accordingly.

) When indicator input is selected

o select indicator input, enter S when the prompt "PERFORMANCE FIGURES=J/

Indicator names to be input

Assets	1. Current assets	8. Sales
	2. Inventories	9. Operating profits
	3. Fixed assets	10. Ordinary profits
	4. Tangible fixed assets	
Liabilities & capital	5. Current liabilities	11. Values added
	6. Equity capital	12. Employees
	7. Total liabilities and net worth	

INDICATOR=S" is displayed. When indicator input is selected, the prompt "INDICA
TOR INPUT CETL=C/KEY=K" is displayed. Now, if you use indicator values whic
have already been prepared by CETL, enter C. If you wish to input the indicato
values from the keyboard, enter K.
When K is entered, the prompt "PERFORMANCE FIGURES=J/INDICATOR=S"

Data table created using CETL

```
File Name:INDICATORS

  Number of Rec.:13
  Number of Item:6


Label Record
Item1:Indication name
   Type(N/S)   :S
   Expression ?
   Format      :&

Item2:Chart name No.1
   Type(N/S)   :N
   Expression ?
   Format      :###,###.##

Item3:Chart name No.2
   Type(N/S)   :N
   Expression ?
   Format      :###,###.##

Item4:Chart name No.3
   Type(N/S)   :N
   Expression ?
   Format      :###,###.##

Item5:Chart name No.4
   Type(N/S)  :N
   Expression ?
   Format      :###,###.##

Item6:Chart name No.5
   Type(N/S)   :N
   Expression ?
   Format      :###,###.##
```

```
Data Area
1-1 :Ratio of ordinary pr
     ofit to total liabil
     ities and net worth
1-2 :  8.55807
1-3 :  9.91306
1-4 :  17.3042
1-5 :  17.762
1-6 :  15.2176

&  2-1 :Ratio of operating p
     rofit to total liabi
     lities and net worth
2-2 :  7.64469
2-3 :  8.46414
2-4 :  15.4783
2-5 :  15.6889
2-6 :  12.6895

3-1 :Ratio of ordinary pr
     ofit to sales
3-2 :  4.18439
3-3 :  4.89467
3-4 :  8.5417
3-5 :  9.14116
3-6 :  8.58655

4-1 :Ratio of operating p
     rofit to sales
4-2 :  3.7378
4-3 :  4.17925
4-4 :  7.64041
```

displayed. Now, if you enter S, the prompt "CHART NO. 2 CHART NAME: _" is displayed, requesting you to input the next chart name. This prompt is repeated until all the chart names are input. After all the chart names are input, input the appropriate indicator values (ratio of ordinary profit to total liabilities and net worth, ratio of operating profit to total liabilities and net worth, . . . , inventory turnover period). When C is entered, the data is input from CETL. In this case, the data table must have been prepared by CETL. Now, when you specify the CETL file area from which the data is to be fetched and the appropriate item number, the prompt "CHART NO. 2 CHART NAME: _" is displayed and repeated until all the chart names are input.
It is of course possible for you to input data for some charts from CETL and data for other charts from the keyboard. In this case, the data input from the keyboard can be added to the data stored in the CETL file (if you like). This is one of the most useful features of the FP-200.

Example of Data Input by CETL

It is recommended that data will be input from a data table created using CETL. For this purpose, first create a data table carrying titles, such as the one shown on page _. Then create an indicator name table (see the same page) from the figures in the title table. By using these data tables, it is possible to obtain the desired charts. As an example, the procedure for preparing an indicator table is shwon below.
Preparing an indicator value table using CETL

Data Input and Output

The program can output a chart of 13 indicators in four different colors, as well as indicator name and title tables. When no title is input, 0 is output in the associated table. For input of performance figures, a routine is provided to permit correcting erroneous data before printout.
As a reference, the equations for calculating the indicators used in this program are given below.

Example of plotter output (when performance figures are input) (in four colors)

CHART FOR BUSINESS ANALYSIS

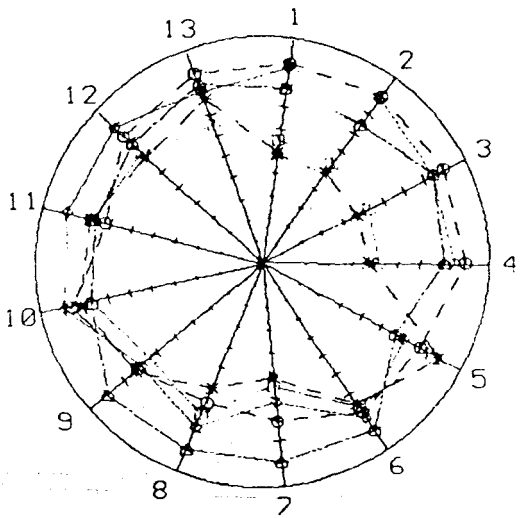


Chart N0. 1
Chart NO. 2
Chart No. 3
Chart No. 4
Chart No. 5

1 Ratio of ordinary profit to total liabilities and net worth
2 Ratio of operating profit to total liabilities and net worth
3 Ratio of ordinary profit to sales
4 Ratio of operating to sales
5 Ratio of fixed assets to net worth
6 Current ratio
7 Ratio of net worth to total liabilities and net worth
8 Sales per employee
9 Value added per employee
10 Value added ratio
11 Turnover ratio of total liabilities and net worth
12 Turnover ratio of tangible fixed assets
13 Inventory turnover period

Indicators

No.	Indicator name	Chart NO.1	Chart NO.2	Chart No.3	Chart No.4	Chart No.5
1	Ratio of ordinary profit to total liabilities and net worth	8.56	9.91	17.30	17.76	15.22
2	Ratio of operating profit to total liabilities and net worth	7.64	8.46	15.48	15.69	12.69
3	Ratio of ordinary profit to sales	4.18	4.89	8.54	9.14	8.59
4	Ratio of operating to sales	3.74	4.18	7.64	8.07	7.16
5	Ratio of fixed assets to net worth	109.96	104.68	84.21	96.44	78.25
6	Current ratio	117.00	120.06	130.34	124.87	141.65
7	Ratio of net worth to total liabilities and net worth	21.16	23.03	26.86	31.21	40.76
8	Sales per employee	17.26	21.65	23.85	19.52	27.55
9	Value added per employee	4.95	4.26	4.65	4.85	6.23
10	Value added ratio	26.25	24.35	28.65	27.32	24.55
11	Turnover ratio of total liabilities and net worth	2.06	2.04	2.36	1.85	1.99
12	Turnover ratio of tangible fixed assets	12.94	13.48	17.26	15.98	14.76
13	Inventory turnover period	16.54	13.58	17.35	19.58	16.22

No.	Item name	Chart No. 1	Chart No 2	Chart No. 3	Chart No. 4	Chart No. 5
1	Current assets	17,726	12,365	12,548	22,654	24,561
2	Inventories	3,402	2,654	2,368	3,248	2,153
3	Fixed assts	5,603	9,653	6,578	9,513	21,053
4	Tangible fixed assets	3,025	2,365	2,698	6,521	5,489

5	Current liabilities	12,563	10,256	15,874	12,395	21,032
6	Equity capital	2,561	1,256	6,541	6,651	10,254
7	Total liabilities and net worth	51,263	32,416	25,987	32,120	15,987
8	Sales	54,123	21,036	65,281	36,525	52,103
9	Operating profit	3,215	1,256	6,123	6,201	6,512
10	Ordinary profit	4,254	3,254	6,542	8,542	4,563
11	Values added	12,036	12,503	12,598	12,690	19,874
12	Employee	4,125	2,365	6,542	3,256	3,210

13 indicators used in this program

Profitability

1. Ratio of ordinary profit to total liabilities and net worth = ordinary profit/total liabilities and net worth x 100 (%)
2. Ratio of operating profit to total liabilities and net worth = operating profit/total liabilities and net worth x 100 (%)
3. Ratio of ordinary profit to sales = ordinary profit/sales x 100 (%)
4. Ratio of operating profit to sales = operating profit/sales x 100 (%)

Liquidity

5. Ratio of fixed assets to net worth = fixed assets/net worth x 100 (%)
6. Current ratio = current assets/current liabilities x 100 (%)
7. Ratio of net worth to total liabilities and net worth = net worth/total liabilities and net worth x 100 (%)

Productivity

8. Sales per employee = sales/total employees
9. Value added per employee = total value added/total employees
10. Value added ratio = value added/sales x 100 (%)

Activity

11. Turnover ratio of total liabilities and net worth = sales/total liabilities and net worth
12. Turnover ratio of tangible fixed assets = sales/tangible fixed assets
13. Inventory turnover period = sales/inventories

LIST

BASIC *2835 Bytes*
CETL *917 Bytes*

```
  100 '
  110 CLEAR 1024:OPTION BASE 1:I=12:J=13:K=5
  120 DIM CN$(K),F(3,K),FI(2,K),J$(I),JS(I,K),MAX(J
),S$(J),SS(J,K)
  130 GOSUB 1040
  140 '
  150 CLS:PRINT T$:PRINT "  Indicators can be read
from or written to CETL.":PRINT "    hit any key";
  160 IF INKEY$="" THEN 160
  170 '========== Specify chart
  180 CLS:PRINT T$:PRINT "Up to five charts can be
plotted. Specify number of charts,chart name,and in
put method."
  190 INPUT "Number of chart :",CH:IF CH<1 OR CH>5
THEN 190
  200 FOR I=1 TO CH:PRINT:PRINT"Chart No.";I
  210 INPUT "Chart name:",CN$(I)
  220 PRINT "Performance figures =J":PRINT "Indicat
or":PRINT "=S"
  230 INPUT " Select (J or S):",Z$:IF Z$="J" THEN F
(1,I)=1 ELSE IF Z$="S" THEN F(1,I)=0 ELSE 230
  240 IF F(1,I)=1 THEN INPUT " Store to CETL ?:",Z$
:IF Z$="Y" THEN F(2,I)=1 ELSE IF Z$="N" THEN F(2,I)
=0 ELSE 240
  250 IF F(1,I)=0 THEN PRINT "Input data":INPUT "CE
TL=C or KEY=K :",Z$:IF Z$="C" THEN F(2,I)=1 ELSE IF
 Z$="K" THEN F(2,I)=0 ELSE 250
  260 IF F(2,I)=1 THEN INPUT "CETL file No.0~9:",Z1
:INPUT"      Item No.2~6:",Z2:IF Z1<0 OR Z1>9 OR Z2<
2 OR Z2>6 THEN 260 ELSE FI(1,I)=Z1:FI(2,I)=Z2
  270 NEXT I
  280 '========== Input performance,indicator
  290 FOR I=1 TO CH:CLS:PRINT CN$(I)
  300 IF F(1,I)=0 THEN Z=13 ELSE Z=12
  310 FOR J=1 TO Z
  320 IF F(1,I)=1 THEN PRINT J$(J):INPUT "-->",JS(J
,I):GOTO 340
  330 PRINT USING U1$;S$(J);:PRINT "-->";:IF F(2,I)
=1 THEN SS(J,I)=FL(FI(1,I),J,FI(2,I)):PRINT SS(J,I)
 ELSE INPUT "",SS(J,I)
```

```
  340 NEXT J,I
  350 '========== Correct performance
  360 FOR I=1 TO CH:IF F(1,I)=1 THEN 370 ELSE NEXT
I:GOTO 420
  370 CLS:PRINT T$:PRINT "Only performance data can
 be corrected."
  380 INPUT "Correct(Y/N):",Z$:IF Z$="N" THEN 420 E
LSE IF Z$<>"Y" THEN 380
  390 INPUT "Chart No. ",Z:IF Z<1 OR Z>CH THEN 390
ELSE IF F(1,Z)=0 THEN 390 ELSE PRINT CN$(Z)
  400 PRINT:INPUT "Performance No.",Z1:IF Z1<1 OR Z
1>12 THEN 400
  410 PRINT J$(Z1):PRINT "  ";JS(Z1,Z);:INPUT "-->"
,JS(Z1,Z):PRINT:GOTO 380
  420 '========== Calculation of indicators
  430 CLS:PRINT T$:PRINT "Being computed."
  440 FOR I=1 TO CH:IF F(1,I)=0 THEN 450 ELSE GOSUB
 860
  450 NEXT I
  460 FOR I=1 TO 13:MAX=-1E99
  470 FOR J=1 TO CH
  480 IF MAX<SS(I,J) THEN MAX=SS(I,J)
  490 NEXT J:MAX(I)=MAX
  500 NEXT I:PRINT " Printing out now !"
  510 '========== Chart
  520 LPRINT CHR$(28);CHR$(37):LPRINT "Y0":LPRINT "
Z2,6":LPRINT "Q0":LPRINT "B3.6":LPRINT "L0":LPRINT
"J0"
  530 LPRINT "S2":LPRINT "PCHART FOR BUSINESS ANALY
SIS":LPRINT "S1"
  540 LPRINT "O50,-45":LPRINT "C0,0,30":ANGLE 0
  550 FOR I=1 TO 13:R=360/13*I-20:C1=30:C2=R:GOSUB
1020:X=CX:Y=CY:C1=33:GOSUB 1020:XM=CX-1:YM=CY-1:Z$=
MID$(STR$(I),2)
  560 IF I>7 THEN XM=XM-1
  570 LPRINT "D0,0,";X;",";Y:LPRINT "M";XM;",";YM:L
PRINT "P";Z$
  580 FOR J=27 TO 3 STEP -3:C1=J:C2=R-2:GOSUB 1020:
X1=CX:Y1=CY:C2=R+2:GOSUB 1020:X2=CX:Y2=CY
  590 LPRINT "D";X1;",";Y1;",";X2;",";Y2
  600 NEXT J,I:LPRINT "S1"
  610 '
  620 FOR I=1 TO CH:LPRINT "L";I MOD 3:LPRINT "J";I
 MOD 4
  630 FOR Z=1 TO 14:K=Z:IF K=14 THEN K=1
```

```
  640 SC=24/MAX(K):C1=SS(K,I)*SC+3:C2=360/13*K-20:G
OSUB 1020:X=CX:Y=CY
  650 IF Z=1 THEN LPRINT "M";X;",";Y ELSE LPRINT "D
,";X;",";Y
  660 LPRINT "N";I+2:NEXT Z,I:LPRINT "H10":LPRINT "
S1"
  670 FOR I=1 TO CH:LPRINT "L";I MOD 3:LPRINT "J";I
 MOD 4
  680 Z=-I*4.8+1:LPRINT "D0,";Z;",10,";Z:LPRINT "N"
;I+2:LPRINT "D,20,";Z:LPRINT "R5,-1":LPRINT "P";CN$
(I)
  690 NEXT I:LPRINT CHR$(28);CHR$(46):LPRINT E$;"S0
":LPRINT E$;"J0":FOR J=1 TO 13:LPRINT USING "## ";J
;:LPRINT S$(J):NEXT J
  700 LPRINT E$;"F5"
  710 '========== Table
  720 IF CH<3 THEN S=1:T0$=TS$ ELSE S=0:T0$=TS$+TS$
  730 LPRINT CHR$(28);CHR$(46):LPRINT E$;"S2":LPRIN
T "Indicators":LPRINT E$;"S";S
  740 FOR L=1 TO 2:IF L=1 THEN Z$="Indicator name"
ELSE Z$="Item name      "
  750 LPRINT "No.";Z$;:LPRINT TAB(25-7*S);:FOR I=1
TO CH:LPRINT USING U1$;CN$(I);:NEXT I:LPRINT
  760 LPRINT T0$:IF L=1 THEN Z1=13 ELSE Z1=12
  770 IF S=0 THEN LPRINT
  780 FOR I=1 TO Z1:IF L=1 THEN Z$=S$(I) ELSE Z$=J$
(I)
  790 IF L=1 THEN N=2 ELSE N=1
  800 LPRINT USING "### ";I:LPRINT E$;"F-1":LPRINT
E$;"S0":LPRINT E$;"F-1":FOR V=0 TO N:Q$=MID$(Z$,20*
V+1,20)
  810 LPRINT TAB(4*S+4);Q$:NEXT V:LPRINT E$;"S0":IF
 N=2 THEN LPRINT E$;"F-2" ELSE LPRINT E$;"F-1"
  820 LPRINT E$;"S";S:LPRINT TAB(25-9*S);:FOR J=1 T
O CH:IF L=1 THEN LPRINT USING "###,###.## ";SS(I,J)
; ELSE LPRINT USING "##,###,### ";JS(I,J);
  830 NEXT J:IF S=1 THEN LPRINT E$;"F1" ELSE LPRINT
 E$;"F3"
  840 LPRINT E$;"S";S:LPRINT :NEXT I:LPRINT T0$:LPR
INT E$;"F5":NEXT L:LPRINT E$;"S1"
  850 GOTO 170
  860 '========== Calculation of indicator
  870 SS(1,I)=JS(10,I)/JS(7,I)*100  'Ratio of ordin
ary profit to total liabilities and net worth
```

```
  880 SS(2,I)=JS(9,I)/JS(7,I)*100    'Ratio of opera
ting profit to total liabilities and net worth
  890 SS(3,I)=JS(10,I)/JS(8,I)*100   'Ratio of ordin
ary profit to sales
  900 SS(4,I)=JS(9,I)/JS(8,I)*100    'Ratio of opera
ting profit to sales
  910 SS(5,I)=JS(3,I)/JS(6,I)*100    'Ratio of fixed
 assets to net worth
  920 SS(6,I)=JS(1,I)/JS(5,I)*100    'Current ratio
  930 SS(7,I)=JS(6,I)/JS(7,I)*100    'Ratio of net w
orth to total liabilities and net worth
  940 SS(8,I)=JS(8,I)/JS(12,I)       'Sales per empl
oyee
  950 SS(9,I)=JS(11,I)/JS(12,I)      'Value added pe
r employee
  960 SS(10,I)=JS(11,I)/JS(8,I)*100 'Value added ra
tio
  970 SS(11,I)=JS(8,I)/JS(7,I)       'Turnover ratio
 of total liabilities and net worth
  980 SS(12,I)=JS(8,I)/JS(4,I)       'Turnover ratio
 of tangible fixed assets
  990 SS(13,I)=JS(8,I)/JS(2,I)       'Inventory turn
over period
 1000 IF F(2,I)=1 THEN FOR J=1 TO 13:FL(FI(1,I),J,F
I(2,I))=SS(J,I):NEXT J
 1010 RETURN
 1020 '========== Calculate X,Y coordinates
 1030 CX=ROUND(C1*COS(90-C2),-3):CY=ROUND(C1*SIN(90
-C2),-3):RETURN
 1040 '========== Set variables
 1050 RESTORE 1100:FOR I=1 TO 12:READ J$(I):FOR J=1
 TO 5:JS(I,J)=0:NEXT J,I
 1060 FOR I=1 TO 13:READ S$(I):FOR J=1 TO 5:SS(I,J)
=0:NEXT J,I
 1070 T$=" *Business analysis*":TS$="":FOR I=1TO 40
:TS$=TS$+"-":NEXT I
 1080 U1$="&          &":E$=CHR$(27)
 1090 RETURN
 1100 '========== Item,indicator name
 1110 DATA Current assets,Inventories,Fixed assts,T
angible fixed assets,Current liabilities,Equity cap
ital,Total liabilities and net worth
 1120 DATA Sales,Operating profits,Ordinary profits
,Values added,Employee
```

```
 1130 DATA Ratio of ordinary profit to total liabil
ities and net worth,Ratio of operating profit to to
tal liabilities and net worth,Ratio of ordinary pro
fit to sales,Ratio of operating to sales
 1140 DATA Ratio of fixed assets to net worth,Corre
nt ratio,Ratio of net worth to total liabilities an
d net worth,Sales per employee,Value added per empl
oyee,Value added ratio,Turnover ratio of total liab
ilities and net worth
 1150 DATA Turnover ratio of tangible fixed assts,I
nventory turnover period
 1160 FOR I=1 TO 13:FL(2,I,1)=S$(I):NEXTI:END
```

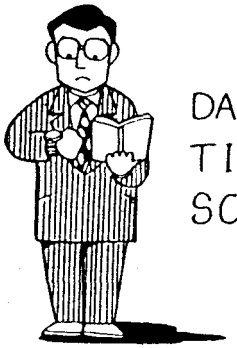
DEBUG

This program permits preparing data tables using CETL. In this case, it is necessary to reserve a CETL area using the AREA command.

SCHEDULING

Program Outline

Perhaps we can remember exactly our schedules for tomorrow, the day after tomorrow, or even a few more days ahead. But it is virtually impossible to remember schedules several weeks or more ahead. In order to avoid forgetting important events, many people use an appointment book in which their schedules are noted.
This program permits using the FP-200 as an 'efficient appointment book' which never forgets.



Data Input and Subtotal Output

When the program is started, it displays the following menu, waiting for data input.

1 New entry
2 Reference (deletion)
3 Additional entry
4 Termination

Input the desired menu number. The contents of the menu are described below.

1. New entry

This is a command to erase all the contents of the previously stored entry and to replace them with a new entry. When you select this menu, input the month, day, time, and the contents of the entry. Note that the month/day and time must be given as 4-digit numbers, as 0228 for February 28, 0503 for 5:03 (a.m.), etc. Failure to do so causes incorrect entry of data. The order in which they are input does not affect the correct input of data. To exit from this routine, input 0 for the month/day data, and the menu is displayed again. A maximum of 80 events can be input.

2. Reference (deletion)

This is a command to check or delete the contents of entries. To check the contents of entries, enter the appropriate date (month/day) as a 4-digit number. If you wish to have a printout of the contents, enter Y (or y) for the prompt. The entries for that date are printed out in the order of the time of occurrence.
Entering anything other than Y (or y) causes the entries to be output on the display in order of the time of occurrence. In this case, to delete the contents being displayed, enter K (or k), and the prompt "DELETE ENTRIES" is displayed for confirmation.
Entering Y (or y)causes the entries to be deleted, and the program waits for another command. Entering anything other than K (or k) while an entry is being displayed causes the next entry to be displayed. After all the entries for that date have been displayed, the program waits for another command.

3. Additional entries

This is a command to insert new entries without erasing the current ones. The same considerations described for new entries are required when using this command. If 80

entries have already been stored, delete the least important entries before adding new ones.

4. Termination

This terminates program execution. A prompt indicating the termination is displayed, and the FP-200 waits for a BASIC command.

It is advisable for the number of entries per day not to exceed 20. If there are more than 20 entries for a particular day, change the figure (20) in the array declaration in line 250 of the program to an appropriate figure. Also, if there are more than 80 entries to be stored, change the figure (80) in the array declaration of line 140 and in the FOR ~ NEXT loop in lines 150, 180, 260, and 430, respectively, to an appropriate figure, and at the same time, change the figure (79) in lines 200 and 220 to the newly assigned figure minus 1. By doing so, the number of entries that can be handled can be increased, although the memory requirement increases accordingly. When either the figure 20 or 80 or both are changed to other figures as described above, re-input all entries using the new entry command.

Example of output

```
DATE    11/16
TIME     10:30
SCHEDULE
MEETING..

DATE    11/24
TIME     18:40
SCHEDULE
PARTY

DATE    11/30
TIME     17:00
SCHEDULE
DATE

DATE    12/05
TIME     19:00
SCHEDULE
BIRTHDAY
```

LIST

1726 Bytes

```
    10 '
    20 'SCHEDULING
    30 '
    40 CLS:PRINT"SCHEDULING"
    50 PRINT"** MENU **"
    60 PRINT"1 NEW ENTRY"
    70 PRINT"2 REFERRENCE":PRINT"     (DELETION)"
    80 PRINT"3 ADDITIONAL ENTRY"
    90 PRINT"4 TERMINATION"
   100 INPUT"SELECT MENU";A
   110 IF A>4 OR A<1 THEN 40
   120 ON A GOTO 130,240,170,570
   130 'NEW ENTRY
   140 CLEAR:DIM SC(80),SC$(80),TI$(80),DE$(80)
   150 CLS:FOR I=0 TO 80
   160 GOSUB 490:NEXT
   170 'ADDITIONAL ENTRY
   180 CLS:FOR I=0 TO 80
   190 IF SC(I)=1 THEN 230
   200 IF I>79 THEN LOCATE 0,3:PRINT"DELETE ANY DATA
  !!":FOR J=0 TO 500:NEXT J:GOTO 40
   210 GOSUB 490
   220 IF I>=79 THEN 40
   230 NEXT I
   240 'REFERENCE
   250 CLS:DIM F(20),TI1$(20),SC1$(20):INPUT"DATE ??
?? ";DT$
   260 K=-1:FOR I=0 TO 80:IF DE$(I)=DT$ THEN K=K+1:F
(K)=VAL(TI$(I)):SC1$(K)=SC$(I)
   270 NEXT I
   280 FOR I=K TO 0 STEP -1:FOR J=0 TO I-1
   290 IF F(J)>F(J+1) THEN SF0=F(J):F(J)=F(J+1):F(J+
1)=SF0:SF1$=SC1$(J):SC1$(J)=SC1$(J+1):SC1$(J+1)=SF1
$
   300 NEXT J,I
   310 PRINT"DO YOU WISH TO OUTPUT TO PRINTER (Y/N)"
   320 Z$=INKEY$:IF Z$="" THEN 320 ELSE IF Z$="Y" OR
 Z$="y" THEN 440
   330 CLS:PRINT"DATE    ";LEFT$(DT$,2);"/";RIGHT$(DT
$,2)
```

```
  340 FOR I=0 TO K:GOSUB 540
  350 PRINT"TIME   ";LEFT$(TI1$(I),3);":";RIGHT$(TI1
$(I),2)
  360 PRINT"SCHEDULE ":PRINT SC1$(I)
  370 Z$=INKEY$:IF Z$="" THEN 370 ELSE IF Z$="K" OR
 Z$="k" THEN 400
  380 PRINT:NEXT I
  390 GOTO 50
  400 'DELETION
  410 PRINT"DO YOU WISH TO DELETE (Y/N)"
  420 Z$=INKEY$:IF Z$="" THEN 420 ELSE IF Z$="Y" OR
 Z$="y" THEN 430 ELSE 380
  430 FOR J=0 TO 80:IF DE$(J)=DT$ AND VAL(TI$(J))=V
AL(TI1$(I)) THEN SC(J)=0:DE$(J)="":GOTO 40 ELSE NEX
T J
  440 'PRINTER
  450 LPRINT"DATE    ";LEFT$(DT$,2);"/";RIGHT$(DT$,2
)
  460 FOR I=0 TO K:GOSUB 540
  470 LPRINT"TIME    ";LEFT$(TI1$(I),3);":";RIGHT$(T
I1$(I),2)
  480 LPRINT"SCHEDULE ":LPRINT SC1$(I):LPRINT:NEXT
I:GOTO 50
  490 'INPUT
  500 INPUT"DATE ???? ";DE$(I)
  510 IF DE$(I)="0" THEN  40
  520 SC(I)=1:INPUT"TIME ???? ";TI$(I)
  530 PRINT"SCHEDULE":INPUT SC$(I):RETURN
  540 'CHARCTER STRING MANIPULATION
  550 TI1$(I)=STR$(F(I)):IF LEN(TI1$(I))=4 THEN TI1
$(I)=" "+TI1$(I) ELSE IF LEN(TI1$(I))=3 THEN TI1$(I
)="   "+TI1$(I)
  560 RETURN
  570 END
```

DEBUG

Key in the program correctly, taking care not to confuse similar variable names (e.g., SC$ with SC1$, TI$ with TI1$, etc.).
If a plotter-printer is not available, lines 310, 320, 440, 450, 460, 470, and 480 need not be input.

INVENTORY CONTROL

This application requires a floppy disk.

Program Outline

For those who sell goods, it is important to record the quantities, amounts, etc., of goods purchased, adjust the inventory amount, and calculate sales and profits using sales data.
This program permits performing inventory control speedily and accurately using a floppy disk. It will prove useful for inventory control operations involving relatively large amounts of data.

Directions

This program requires that a data file have been previously created on a floppy disk.
First, execute the program in LIST 1. When the program displays the prompt "FILE NAME," input the file name, and the associated file will be created on the floppy disk. Then, execute the program in LIST 2. When the prompt "DISK SET OK? (Y/N)" is displayed, check that the floppy disk is ready and press the RETURN key. The screen then displays all the file names on the floppy disk. Inputting the previously entered file name then causes the menu (as shown below) to be displayed.
There are 13 controlled items, four of which (quantity of stock, amount of stock, amount of sales, profit) are calculated automatically.

❶ Menu

```
1) DATA ALL PRINT
2) TOTAL PRINT
3) DATA EDIT
4) DATA ADD
5) DATA KILL
6) DATA SAVE
7) E N D
Selection ?
```

❷ Controlled Items

1) Product Name
2) Classification
3) Memo
4) Purchase price
5) Selling price A
6) Selling price B
7) Quantity purchased
8) Quantity sold
9) Quantity on hand
10) Value of goods on hand
11) Sales Amount
12) Profit
13) Remarks

Each of the operations shown in the menu is performed by inputting the associated number. When inputting data for the first time, specify 4) Data Add.

(1) Print All Data

The data about all products stored in the file is displayed on the screen. The data is sequentially displayed starting from the product name. Press any key other than the E key to display the next item.
Pressing the E key causes the program to return to the menu screen.

(2) Print Total

The total amount of each of the products is displayed. The items displayed are from 4) purchase price to 12) profit (see 'Controlled Items'). The procedures for displaying individual items and returning to the menu screen are the same as described in (1) above.

(3) Edit Data

In this mode, the contents of stored product data can be modified. To enter this mode, press other than the E or N key when the program displays 'EDIT DATA.' Data is arranged in horizontal rows, and the products are arranged in a vertical column. To find the data to be changed, use the CURSOR key. Pressing the CURSOR key causes a shift by one item. The shift can be made in any of four directions. When the cursor reaches either end, it moves in the opposite direction.

When the desired item is found, press the E key, and the program displays "NEW DATA." Input the new data. After changing all the relevant data, press the N key and the program returns to the menu screen.

(4) Add Data

In this mode, new product data can be added. Input the product name, classification, memo, purchase price, selling price A, selling price B, quantity purchased, and quantity sold. Then, the program displays the calculated quantity on hand, value of goods on hand, sales and profits. Finally, the input of remarks terminates data input for one product. The program now displays "OK? (Y/N)/END (E)." Check the data just input. If it is correct, press any key except the E or N key to input the next data. If an error is found, press the N key and correct the error. At the end of data input, press the E key to return to the menu screen.

(5) Kill Data

This mode is used to delete unnecessary product data. Input the data number to be deleted, and the contents of four items are displayed. Check the contents and press the Y key. Pressing anything except the Y key simply causes a return to the menu screen.

(6) Save Data

In this mode, the contents of a file modified by steps (3), (4) or (5) are saved on the floppy disk. The program displays all the file names on the floppy disk, asking for the appropriate file name.If you wish to rewrite the original file, input the file name. If you wish to save data to another file, input a new file name.

(7) End

Terminates program execution. Note that when data has been changed, Save Data must be excecuted before End.

LIST

LIST 1 *84 Bytes*
LIST 2 *4202 Bytes*

LIST 1

```
   10 INPUT "FILE NAME ";N$
   20 OPEN N$ FOR OUTPUT AS #1
   30 FOR A=1 TO 10
   40 PRINT #1,"*":NEXT
   50 CLOSE
```

LIST 2

```
  100 CLEAR:CLS:PRINT "* INVETORY CONTROL *"
  110 INPUT "DISK SET OK? [Y/N]";A$:IF A$="N" THEN
END
  120 FILES:INPUT "[FILE NAME]";N$:OPEN N$ FOR INPU
T AS #1
  130 DIM D$(50,12)
  140 FOR I=0 TO 12:READ D$(0,I):NEXT
  150 DATA Product name,Classification,Memo,Purchas
e price,Selling price A,Selling price B,Quantity pu
rchased,Quantity sold,Quantity on hand,Value of goo
ds on hand,Sales amount,Profit,Remarks
  160 A=1
  170 FOR I=0 TO 12:IF EOF(1)=-1 THEN CLOSE:GOTO 20
0
  180 INPUT #1,D$(A,I)
  190 NEXT:A=A+1:GOTO 170
  200 A=A-1:PRINT "Number of data";A:FOR I=0 TO 300
:NEXT
  210 CLS:PRINT "MENU"
  220 PRINT "1) DATA ALL PRINT"
  230 PRINT "2) TOTAL PRINT"
  240 PRINT "3) DATA EDIT "
  250 PRINT "4) DATA ADD"
  260 PRINT "5) DATA KILL"
  270 PRINT "6) DATA SAVE"
  280 PRINT "7) E N D"
  290 INPUT "Selection";B$:B=VAL(B$):IF B<1 OR B>7
THEN 210
  300 ON B GOTO320,400,460,580,680,760,800
  310 GOTO 290
  320 CLS:PRINT "DATA ALL PRINT "
  330 FOR I=1 TO A:PRINT USING "----- <<No-##>> -----
";I
```

```
  340 FOR J=0 TO 12
  350 PRINT "[";D$(0,J);"]"
  360 PRINT "  ";D$(I,J)
  370 A$=INKEY$
  380 IF A$="" THEN 370 ELSE IF A$="E" OR A$="e" TH
EN 210
  390 NEXT :NEXT:GOTO 210
  400 CLS:PRINT "TOTAL PRINT":PRINT "---------------
-----"
  410 FOR I=3 TO 11:S#=0:FOR J=1 TO A:S#=S#+VAL(D$(
J,I)):NEXT
  420 PRINT "[";D$(0,I);"]":PRINT S#
  430 IF INKEY$="" THEN 430
  440 NEXT
  450 GOTO 210
  460 CLS:PRINT "DATA EDIT"
  470 X=1:Y=0
  480 A$=INKEY$:IF A$="" THEN 480
  490 IF A$="N" OR A$="n" THEN 210
  500 IF A$=CHR$(30) THEN X=X-1:IF X<1 THEN X=A
  510 IF A$=CHR$(31) THEN X=X+1:IF X>A THEN X=1
  520 IF A$=CHR$(29) THEN Y=Y-1:IF Y<0 THEN Y=12
  530 IF A$=CHR$(28) THEN Y=Y+1:IF Y>12 THEN Y=0
  540 IF A$="E" OR A$="e" THEN GOSUB 570:PRINT D$(X
,Y):INPUT "NEW DATA ";D$(X,Y)
  550 GOSUB 570:PRINT D$(X,Y)
  560 GOTO 480
  570 CLS:PRINT "No.";X:PRINT "[";D$(0,Y);"]":PRINT
 "--------------------":F=1:GOSUB 600:RETURN
  580 CLS:PRINT "DATA ADD "
  590 A=A+1:PRINT USING "---- <<No-##>> ----";A:FOR
 I=0 TO 7:PRINT "[";D$(0,I);"]":INPUT "  ";D$(A,I):
NEXT:F=0:GOSUB 600:GOTO 630
  600 IF F=0 THEN Z=A ELSE Z=X
  610 C#=VAL(D$(Z,6))-VAL(D$(Z,7)):D$(Z,8)=STR$(C#)
:C#=(VAL(D$(Z,6))-VAL(D$(Z,7)))*VAL(D$(Z,3)):D$(Z,9
)=STR$(C#)
  620 C#=VAL(D$(Z,4))*VAL(D$(Z,7)):D$(Z,10)=STR$(C#
):C#=VAL(D$(Z,10))-VAL(D$(Z,3))*VAL(D$(Z,7)):D$(Z,1
1)=STR$(C#):RETURN
  630 FOR I=8 TO 11:PRINT "[";D$(0,I);"]":PRINT "
";D$(A,I):FOR W=1 TO 200:NEXT:NEXT
  640 PRINT "[";D$(0,12);"]":INPUT "  ";D$(A,12)
  650 INPUT "OK ?[Y/N] / END[E]",Y$
```

```
  660 IF Y$="E" OR Y$="e" THEN 210 ELSE IF Y$<>"N"
THEN 590
  670 A=A-1:GOTO 590
  680 CLS:PRINT "DATA KILL":PRINT "----------------
----"
  690 PRINT "DATA ";A;"Aitems"
  700 INPUT "Which data";U
  710 IF U<1 OR U>A THEN 210 ELSE FOR I=0 TO 3:PRIN
T "  ";D$(U,I):NEXT:PRINT "OK? [Y/N]";
  720 Y$=INKEY$:IF Y$="" THEN 720
  730 IF Y$<>"Y" THEN 210
  740 A=A-1:FOR I=U TO A
  750 FOR J=0 TO 12:D$(I,J)=D$(I+1,J):NEXT:NEXT:GOT
O 210
  760 CLS:PRINT "DATA SAVE ":FOR T=0 TO 300:NEXT
  770 FILES:INPUT "[FILE NAME]",F$:OPEN F$ FOR OUTP
UT AS #1
  780 FOR I=0 TO A
  790 FOR J=0 TO 12:PRINT #1,D$(I,J):NEXT:NEXT:CLOS
E:GOTO 210
  800 END
```

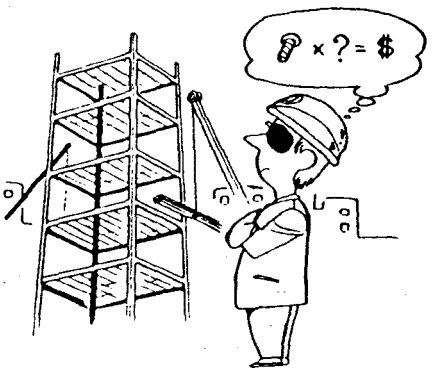
DEBUG

Before executing this program, be sure to execute MOUNT 1. Once executed, it is effective even after the power is turned off.

ESTIMATION

Program Outline

Preparing an estimate for a particular commodity or job is time-consuming if the unit price of each of many items involved has to be checked by referring to a price list.
This program permits preparing an estimate quickly merely by inputting the quantity and discount rate of a particular commodity, which can be easily retrieved from a previously stored list. Since commodities are recorded under CETL, the updating of data and the transfer of data to or from an external device are easy. This program will prove useful in transactions with customers and in preparing estimates at shops.



Directions

When the program is started, a menu is displayed on the screen. The following describes the contents of the menu and how to use the menu.

(1) Command 1: Storing product data

When command 1 is selected, the prompt "PRODUCT NAME (MAX 9 CHARS)" is displayed. Input the product name (of 9 characters or less) and press the ↵ key. If a product name exceeding 9 characters in length is input, the prompt is displayed again, asking for a shorter product names.
The prompt "UNIT PRICE" is then displayed. Input the unit price of the product and press the ↵ key. The prompt "OK (Y/N)" will be displayed. If you wish to correct the unit price just input, press the N key and correct the input data. If there is no need to correct the input data, press the Y key.
The product name and the unit price are automatically filed in CETL. When the CETL file becomes full of input data, no more data is accepted. In this case, the message "CETL INPUT OVER" is displayed and the program returns to the menu.
After the product name and the unit price are input, the prompt "END (E/ELSE)" is displayed. To terminate the input, press the E key. To continue data input, press any key except the E key. Pressing the E key causes a return to the menu.

(2) Command 2: Preparing an estimate

When command 2 is selected, the program asks for input of the product name. Input the product name to be retrieved and press the ↵ key. Then, the message "SEARCH!!" is displayed, indicating that the product name is being searched for. When the product name is found, the message "FOUND!!" is displayed, followed by the product name and the unit price. Here, input the quantity and the discount rate for the product. The program then displays the calculated amount for quotation.
If the specified product name is not found, the program displays the message "NOT FOUND" and enters data entry mode (command 1). In this case, the program asks

whether or not data is to be input to CETL. Press the Y key (if data is to be input) or the N key (if not), and then press the ↵ key. When the appropriate data is input, the program returns to command 2.

After displaying the calculated amount, the program displays the prompt "END (Y/N/C)." Press the Y key to terminate command 2; press the N key to calculate an estimate for another product; press the C key to calculate an estimate for the same product using a different quantity and discount rate. The program then asks whether or not all the data is to be displayed. Pressing the Y key and the ↵ key causes all the data to be displayed.

The program next asks whether or not the total amount is to be calculated. Pressing the Y key causes the program to start calculating the total quantity, total amount, and total selling price (i.e., total amount minus total discount). Pressing the ↵ key when the totals are displayed causes a return to the menu.

(3) Command 3: Listing products

When command 3 is selected, the prompt "SCREEN (1) OR PRINTER (2)" is displayed. Key in 1 or 2, and press the ↵ key.

If 1 (screen) is keyed in, seven product names and associated unit prices are displayed on the screen at a time. To display the next seven product names and unit prices, press the ↵ key.

After a list of the products is output, the program returns to the menu.

Of the 16 CETL commands, the following commands can be effectively used for this program:

D (Delete Data)

Delete part or all of the data. Since this program does not have a delete command, use the CETL command to delete data. Note that the B (Blank) command must not be used to delete data.

S (Sort)

Sort data. For the prompt "KEY ITEM?," input either 1 (product name) or 2 (unit price). For the prompt "REC.?," input 1, XX. XX indicates the last serial number displayed by command 3 of this program.

F (Find)

Search for specified data. For the prompt "REC.?," input 1, XX as for the S command. For the prompt "CONDITION?," input IT(2)<500, for example. In this case, products whose unit prices are lower than $500 (the content of item 2) are searched for.

T (Table)

This command has the same function as command 3 of the program.

P (Put)

Output data to a floppy disk or cassette tape.

G (Get)

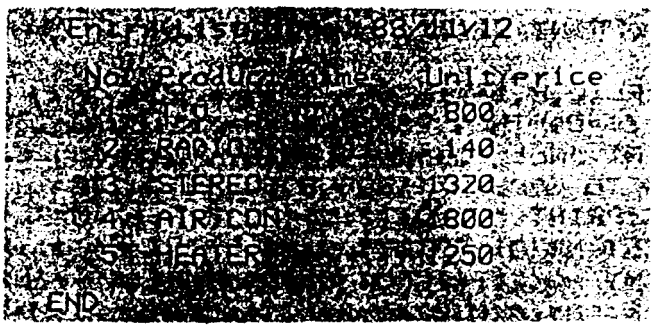
Input data from a floppy disk or cassette tape.

Note on inputting the program

When inputting the BASIC program, be sure to include the REM statements. Otherwise a UL error will occur due to the missing destination line number for a GOSUB.

When creating a CETL file, use the N command to input the file name, the number of

records and the number of items. Next, input the type, expression and format of each item. For the expression, just press the ↵ key. After inputting the contents of items, press the BREAK key. There is no need to input any data.



LIST *2803 Bytes*

```
   10 REM *** INIT & MAIN ***
   20 CLEAR:DIM N$(5),A$(50),A(50,4)
   30 FOR I=0 TO 5:READ N$(I):NEXT
   40  DATA Product name,Unit price,Quantity,Amount
,Discount rate,Selling price
   50 CLS:PRINT "** ESTIMATE MENU **":PRINT :PRINT
"  1:Entry":PRINT "  2:Estimate":PRINT "  3:Entry L
IST":PRINT "  4:END":PRINT :INPUT"Which command?";Z
   60 Y=0:ON Z GOSUB 80,200,530,720
   70 GOTO 50
   80 REM *** Entry ***
   90 CLS
  100 PRINT N$(0);"          (MAX 9 CHARS)":INPUT M$:
IF LEN(M$)>9 THEN 100 ELSE PRINT N$(1);:INPUT M
  110 PRINT "OK(Y or N)?"
  120 Z$=INKEY$:IF Z$="Y" THEN 130 ELSE IF Z$="N" T
HEN 100 ELSE120
  130 IF Y THEN INPUT"Do you wish to input data to
CETL?";Z$:IF Z$="N" THEN RETURN
  140 P=1:PRINT "Input data to CETL"
  150 IF FL(0,P,1)<>"" THEN P=P+1:IF P>500 THEN PRI
NT "CETL INPUT OVER!!":FOR I=0 TO 300:NEXT:RETURN:E
LSE 150
  160 FL(0,P,1)=M$:FL(0,P,2)=M
  170 IF Y THEN RETURN
  180 PRINT "END(E or ELSE)?"
```

```
  190 Z$=INKEY$:IF Z$="E" THEN RETURN ELSE IF Z$<>"
" THEN 90 ELSE 190
  200 REM *** Estimate ***
  210 K=0:CLS
  220 P=1:PRINT N$(0);" Input..":INPUT Z$:PRINT "SE
ARCH!!"
  230 O$=FL(0,P,1):IF O$="" THEN 250
  240 IF Z$=O$ THEN 260 ELSE P=P+1:GOTO 230
  250 PRINT "Not found!!":PRINT "Catalog data ":Y=-
1:FOR I=0 TO 300:NEXT:GOSUB 80:GOTO 270
  260 CLS:PRINT "FIND!!":M$=FL(0,P,1):M=FL(0,P,2)
  270 PRINT N$(0):PRINT" ";M$:PRINT N$(1):PRINTM
  280 PRINT N$(2);:INPUT U:PRINT N$(4);:INPUT V
  290 A$(K)=M$:A(K,0)=M:A(K,1)=U:A(K,2)=M*U:A(K,3)=
V:A(K,4)=(1-V/100)*A(K,2)
  300 GOSUB 460
  310 PRINT "END(Y or N or C)?"
  320 Z$=INKEY$:IF Z$="Y" THEN 330 ELSE IF Z$="N" T
HEN K=K+1:GOTO 220 ELSE IF Z$="C" THEN 280 ELSE 320
  330 PRINT "Do you wish to      display all data?"
  340 Z$=INKEY$:IF Z$="Y" THEN 350 ELSE IF Z$="N" T
HEN 400 ELSE 340
  350 I=K:FOR K=0 TO I
  360 GOSUB 460
  370 PRINT "PUSH RETURN KEY      TO CONTINUE";
  380 IF INKEY$<>CHR$(13) THEN 380
  390 NEXT
  400 PRINT :PRINT "Do you wish to have total?"
  410 Z$=INKEY$:IF Z$="Y" THEN 420 ELSE IF Z$="N" T
HEN RETURN ELSE 410
  420 PRINT "Being computed":K=K+1:FOR I=0 TO K-1:F
OR J=0 TO I:A(K,2^J)=A(K,2^J)+A(I,2^J):NEXT:NEXT
  430 CLS:Z$=" Total":PRINT N$(2);Z$:PRINTA(K,1):PR
INT N$(3);Z$:PRINTA(K,2):PRINT N$(5);Z$:PRINTA(K,4)
  440 PRINT "PUSH RETURN KEY                    TO MENU
";
  450 IF INKEY$<>CHR$(13) THEN 450 ELSE RETURN
  460 CLS:PRINT N$(0);" :";A$(K):FOR I=0 TO 4:PRINT
 N$(I+1)
  470 PRINTA(K,I)
  480 IF I=2 THEN GOSUB 500
  490 NEXT:RETURN
  500 PRINT"<< Press any key >>";
  510 IF INKEY$="" THEN 510
  520 CLS:RETURN
```

```
  530 REM *** Entry List ***
  540 INPUT"Screen (1) or        Printer(2)";Z
  550 IF Z<>1 AND Z<>2 THEN 530
  560 P=1:ON Z GOTO 570,660
  570 PRINT "Press RETURN key andproceed to next st
ep"
  580 IF INKEY$<>CHR$(13) THEN 580
  590 CLS:I=1
  600 IF FL(0,P,1)="" THEN 640
  610 PRINT USING"### &        &#######";P,FL(0,P,1),
FL(0,P,2)
  620 P=P+1:I=I+1:IF I<8 THEN 600
  630 IF INKEY$=CHR$(13) THEN 590 ELSE 630
  640 PRINT "END":PRINT"(PUSH RETURN KEY)"
  650 IF INKEY$=CHR$(13) THEN RETURN ELSE 650
  660 PRINT "Connect printer      on-line and press
  RETURN key"
  670 IF INKEY$<>CHR$(13) THEN 670
  680 LPRINT :LPRINT " Entry List        ";DATE$:LPR
INT:LPRINT "  No. Product name   Unit price"
  690 IF FL(0,P,1)="" THEN 710
  700 LPRINT USING"####  &        &     ######";P,FL(
0,P,1),FL(0,P,2):P=P+1:GOTO 690
  710 LPRINT :LPRINT "END":RETURN
  720 REM *** END ***
  730 END
```

Input data stored in CETL

```
File Name:ESTIMATE

     Number of Rec.:3
     Number of Item:2


Label Record
Item1:PRODUCT
   Type(N/S)   :N
   Expression ?
   Format      :&       &

Item2:UNIT
   Type(N/S)   :N
   Expression ?
   Format      :######
```

DEBUG

Change the program

When using this program, the user will have to change the following two points:

(1) Amount of data

Change the number 500 in the IF statement (P>500) in line 170 to the actual number of data items. The number of records for the preparation of a CETL file should also be changed to the actual number of data items + 1.

The smaller the amount of data, the higher the speed of input/output operation, especially when a cassette tape is involved.

(2) File number

If file 1 is already occupied, create the file in another file area using CETL and change the BASIC program as follows:

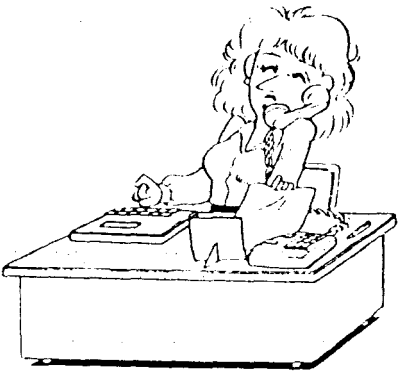
Assign the file number actually used to the first parameter of the FL functions in lines 170, 180, 330, 360, 670, 680, 760, and 770.

CUSTOMER MANAGEMENT

This application requires a floppy disk and printer.

Program Outline

In dealing with customers, it is very important to record their names, addresses, telephone numbers, etc. accurately.
This program is intended for effective customer management. It permits handling the name, postal code, address, phone number, birthday, and identification number, as well as three reference items, for each customer. A floppy disk used for filing the records permits high-speed transfer of large volumes of data.
The program also provides the capability of printing out a list of names and addresses for direct mail operations.



Directions

Before executing this program, it is necessary to create a data file on the floppy disk using the program shown in listing 1. When the program is executed, it asks for the file name. Input the appropriate file name.
Next, execute the program in listing 2. When the prompt "DISK SET OK? (Y/N)" is displayed, check that the floppy disk is ready and press the RETURN key. The program then displays all the file names on the floppy disk, and asks for the file name to be used. Input the previously entered file name. When the file is loaded into memory, the program displays the number of data items, then returns to menu ❶.
This menu can be used to perform ordinary operations. A submenu (menu❷) is also available (specify (7) in menu 1). When entering data for the first time, select (4) Data Add in menu 1.

❶ Menu 1

```
MENU 1
1) PRINT ALL DATA
2) SORT DATA
3) EDIT DATA
4) ADD DATA
5) KILL DATA
6) SAVE DATA
7) MENU 2
```

❷ Menu 2

```
MENU 2
1) Retrieve name
2) Tabulate
3) Print address
4) Enter new item
5) MENU 1
6) END
```

Menu 1

(1) Print All Data

All the customer data stored in the file is displayed item by item. The data items are displayed in the order in which they were entered. To move from one item to another, press other than the E key.

After all the data is displayed, the program returns to the menu. Pressing the E key also causes the program to return to the menu even if the display of data is not completed.

(2) Sort Data

The data can be sorted in character code order using any item as the key. By specifying an appropriate item number (1 for name, 2 for postal code, etc.) in response to the prompt "ITEM NO. (1–9)," it is possible to sort all the data according to the item. After the sort operation is completed, the program returns to the menu. Pressing any key other than keys 1–9 causes the program to return to the menu without performing the sort operation.

(3) Edit Data

Any data item can be changed. First, the program displays the message "EDIT DATA." Pressing any key other than E or N causes the program to enter edit mode. Items are arranged in horizontal rows and records are arranged in a vertical column. Use the cursor keys to find the data to be changed. Any of the cursor keys causes movement of one item each time it is pressed. After the cursor reaches one end, it moves in the opposite direction when it is next pressed. When the data to be changed is found, press the E key and the program displays "NEW DAT." Input the new data to replace the original data. Any data can be changed in this way. After changing the data, press the N key to return to the menu.

(4) Add Data

In this mode, new data can be added. The program displays the number of items and the item names sequentially. After you input a customer's data, the program displays the prompt "OK? (Y/N)/END (E)". If the input data is correct, press any key except the E key. Continue inputting data in this manner. To return to the menu, press the E key.

(5) Kill Data

This is used to delete unnecessary data. The program first asks for the data number to be deleted. Input the appropriate data number, and the program displays the contents of the data. Check that the data is the one which is to be deleted and press the Y key. Pressing any key except the Y key causes the program to return to the menu without deleting the data.

(6) Save Data

This is used to update a file on a floppy disk when the contents of that file are modified.

The program displays all the file names on the floppy disk and asks for the file name to be saved.

Input the file name that was loaded into memory when the program was executed; the file will be updated. If any other file name is input, that file is created as a new file.

(7) Menu 2

The program moves to menu 2.

Menu 2

(1) Retrieve Name

Retrieves data about a specified name from the file. Input an appropriate name when the program displays the prompt "NAME?". The program then searches the file for the associated data and displays part of the data. If it is part of the desired data, press any key except the N key and the program will display the data. If it is irrelevant to the desired data, press the N key to continue the retrieve operation. If the data for the specified name is not found, the program displays "DATA NOT FOUND." In this case, press the RETURN key, and the program returns to the menu.

(2) Tabulate

In this mode, all the data in a file can be output on the printer. When the program displays the prompt "PRINTER SET? (Y/N)," check that the printer is ready, and press the RETURN key. The printer prints the item names, followed by all the associated data. At the end of the printout the program returns to the menu.

(3) Print Address

In this mode, names and addresses can be printed for direct mailing. If you wish to print a specific name and address, press the T key. If you wish to print all names and addresses, press the A key. When you press the T key, the program displays the prompt "WHICH NAME." Specify the name by its associated number in the table obtained by (2) Tabulate. After the name and address are printed, the program automatically returns to the menu. If labels are to be printed, change the TAB and LPRINT statements in the program according to the appropriate print width.

(4) Entry new item

This program provides three extra items for reference information for the user. In this mode, the program displays the current item name (the initial value is *). Specify the item number (6–8) to be entered and the program will display the prompt "NAME?". Input the item name. After one item has been entered the program returns to the menu.

(5) Menu 1

The program displays menu 1.

(6) End

Terminates program execution. Note that when data is changed, a Save Data must be executed before terminating the program.

LIST

LIST 1 *84 Bytes*
LIST 2 *4916 Bytes*

LIST 1

```
10 INPUT "FILE NAME ";N$
20 OPEN N$ FOR OUTPUT AS #1
30 FOR A=1 TO 10
40 PRINT #1,"*":NEXT
50 CLOSE
```

Only necessary for creating file for the first-time

LIST 2

```
  100 CLEAR:CLS:PRINT "*** CUSTOMER                    MAN
AGEMENT ***":PRINT
  110 INPUT "DISK SET OK? [Y/N]",A$:IF A$="N" THEN
END
  120 FILES:INPUT "[FILE NAME]",N$:OPEN N$ FOR INPU
T AS #1
  130 DIM D$(50,8)
  140 FOR I=0 TO 5:READ D$(0,I):NEXT
  150 DATA Name,Postal cord,Address,Telephone No,Bi
rthday,ID No
  160 A=1:FOR I=6 TO 8:INPUT #1,D$(0,I):NEXT
  170 FOR I=0 TO 8:IF EOF(1)=-1 THEN CLOSE :GOTO 20
0
  180 INPUT #1,D$(A,I)
  190 NEXT:A=A+1:GOTO 170
  200 A=A-1:PRINT "Number of data";A:FOR I=0 TO 300
:NEXT
  210 CLS:PRINT "MENU 1"
  220 PRINT "1) PRINT ALL DATA"
  230 PRINT "2) SORT DATA"
  240 PRINT "3) EDIT DATA"
  250 PRINT "4) ADD  DATA"
  260 PRINT "5) KILL DATA"
  270 PRINT "6) SAVE DATA"
  280 PRINT "7) MENU 2";
  290 INPUT B$:B=VAL(B$):IF B<1 OR B>7 THEN 210
  300 ON B GOTO 310,380,440,560,620,710,750
  310 CLS:PRINT "DATA ALL PRINT "
  320 FOR I=1 TO A:PRINT USING "----- <<No-##>> ----
";I
  330 FOR J=0 TO 8
  340 PRINT "[";D$(0,J);"]":PRINT "   ";D$(I,J)
  350 A$=INKEY$
  360 IF A$="" THEN 350 ELSE IF A$="E" OR A$="e" TH
EN 210
  370 NEXT:NEXT:GOTO 210
  380 CLS:PRINT "DATA SORT"
  390 INPUT "ITEM NO.(1-9)";O:O=O-1:IF O<0 OR O>8 T
HEN 390
  400 FOR I=1 TO A-1:FOR J=I+1 TO A
  410 IF D$(I,O)>D$(J,O) THEN FOR K=0 TO 8:A$=D$(I,
K):D$(I,K)=D$(J,K):D$(J,K)=A$:NEXT
  420 NEXT:NEXT
  430 GOTO 210
```

```
  440 CLS:PRINT "DATA EDIT"
  450 X=1:Y=0
  460 A$=INKEY$:IF A$="" THEN 460
  470 IF A$="N" OR A$="n" THEN 210
  480 IF A$=CHR$(30) THEN X=X-1:IF X<1 THEN X=A
  490 IF A$=CHR$(31) THEN X=X+1:IF X>A THEN X=1
  500 IF A$=CHR$(29) THEN Y=Y-1:IF Y<0 THEN Y=8
  510 IF A$=CHR$(28) THEN Y=Y+1:IF Y>8 THEN Y=0
  520 IF A$="E" OR A$="e" THEN GOSUB 550:PRINT D$(X
,Y):INPUT "NEW DATA ";D$(X,Y)
  530 GOSUB 550:PRINT D$(X,Y)
  540 GOTO 460
  550 CLS:PRINT "No.";X;"[";D$(0,Y);"]":PRINT "-----
----------------":RETURN
  560 CLS:PRINT "ADD DATA"
  570 A=A+1:PRINT USING "----- <<No-##>> -----";A
  580 FOR I=0 TO 8:PRINT "[";D$(0,I);"]":INPUT "  "
;D$(A,I):NEXT
  590 INPUT "OK ?[Y/N] / END[E]",Y$
  600 IF Y$="E" OR Y$="e" THEN 210 ELSE IF Y$<>"N"
THEN 570
  610 A=A-1:GOTO 570
  620 CLS:PRINT "DATA KILL":PRINT "-----------------
---"
  630 PRINT "Number of data";A
  640 INPUT "Which data";U
  650 IF U<1 OR U>A THEN 210 ELSE FOR I=0 TO 3:PRIN
T "  ";D$(U,I):NEXT:PRINT "OK ? [Y/N]";
  660 Y$=INKEY$:IF Y$="" THEN 660
  670 IF Y$<>"Y" THEN 210
  680 A=A-1
  690 FOR I=U TO A
  700 FOR J=0 TO 8:D$(I,J)=D$(I+1,J):NEXT:NEXT:GOTO
 210
  710 CLS:PRINT "DATA SAVE ":FOR T=0 TO 200:NEXT
  720 FILES:INPUT "[FILE NAME]",F$:OPEN F$ FOR OUTP
UT AS #1:FOR I=6 TO 8:PRINT #1,D$(0,I):NEXT
  730 FOR I=1 TO A
  740 FOR J=0 TO 8:PRINT #1,D$(I,J):NEXT:NEXT:CLOSE
:GOTO 210
  750 CLS:PRINT "MENU 2"
  760 PRINT "1) Retrieve name"
  770 PRINT "2) Tabulate"
  780 PRINT "3) Print address"
  790 PRINT "4) Enter new item"
```

```
  800 PRINT "5) MENU 1"
  810 PRINT "6) END"
  820 INPUT "Select";B$:B=VAL(B$):IF B<1 OR B>6 THE
N 750
  830 ON B GOTO 840,910,950,1070,210,1120
  840 CLS:PRINT "Retrieve name":PRINT "----------
--------":INPUT "[NAME]",F$
  850 FOR I=1 TO A:IF LEFT$(D$(I,0),LEN(F$))=F$ THE
N FOR J=0 TO 4:PRINT "  ";D$(I,J):NEXT:INPUT "OK";B
$:IF B$="N" THEN 860 ELSE O=I:GOTO 880
  860 NEXT
  870 PRINT "Data not found":INPUT "  [PUSH RETKEY]
",A$:GOTO 750
  880 CLS:FOR I=0 TO 8:PRINT "[";D$(0,I);"]":PRINT
"  ";D$(O,I)
  890 IF INKEY$="" THEN 890 ELSE NEXT
  900 GOTO 750
  910 CLS:PRINT "Tabulate":PRINT:INPUT "PRINTER SET
 [Y/N]",A$:IF A$="N" OR A$="n" THEN 750
  920 PRINT "Being printed":FOR I=0 TO A:IF I<>0 TH
EN LPRINT:LPRINT"<No.";I;">"
  930 FOR J=0 TO 8
  940 LPRINT USING"&                    & ";D$(I,J);:NEXT:
LPRINT:NEXT:GOTO 750
  950 CLS:PRINT "Print address"
  960 PRINT:PRINT "  All print     (A)":PRINT "  Sel
ect print (T)":PRINT "  Cancel        (N)"
  970 I$=INKEY$:IF I$="" THEN 970 ELSE IF I$="N" TH
EN 750
  980 IF I$="A" THEN 1010 ELSE IF I$="T" THEN INPUT
 "Which name";I ELSE 750
  990 IF I<1 OR I>A THEN 750
 1000 GOSUB 1020 :GOTO 750
 1010 FOR I=1 TO A:GOSUB 1020:NEXT:GOTO 750
 1020 LPRINT TAB(5);D$(I,1)
 1030 LPRINT TAB(2);D$(I,2)
 1040 LPRINT TAB(5);D$(I,0)
 1050 LPRINT TAB(9);"TEL ";D$(I,3)
 1060 LPRINT:LPRINT:RETURN
 1070 CLS:PRINT "Catalog items"
 1080 LOCATE 0,1:FOR I=6 TO 8:PRINT I;D$(0,I):NEXT
 1090 INPUT "NO.";I$:IF I$="6" OR I$="7" OR I$="8"
THEN I=VAL(I$) ELSE 750
 1100 INPUT "NAME ";D$(0,I)
 1110 GOTO 750
 1120 END
```

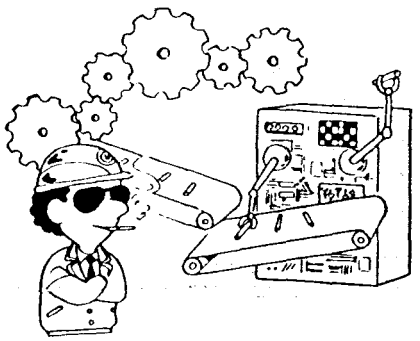
$\bar{x}$-R CONTROL CHART

This application requires a plotter-printer.

Program Outline

Control charts are used mainly for quality control and analysis of processes. They are useful for obtaining an overall view of particular data. The $\bar{x}-R$ chart is typical of various control charts based on measured values. The $\bar{x}-$ chart represents the change in the mean value in the distribution of data and the R-chart represents the change in the range of the data.

Input data can be preserved on cassette tape. The results of processing are output in the form of charts (graphs) and a table. This application requires the FP-1011PL plotter-printer.



Directions

A Chart preparation process

1) Collected data is arranged according to the time of measurement or lot number, then divided into groups, which are input one at a time. The size n of a group (the number of data items comprising a group) is from 2 to 6. The value of n should be predetermined.
2) For each group, the mean value $\bar{x}$ and the range R are calculated.
   Mean value $\bar{x} = \Sigma x/n$
   Range $R = x_{max} - x_{min}$ (x_{max} : maximum value, x_{min} : minimum value);
3) The total mean and the average range are calculated.
   Total mean $\bar{\bar{x}} = \Sigma x/k$
   Averange range $\bar{R} = \Sigma R/k$
   k: Number of groups.
4) The control limits for $\bar{x}$ and R charts are calculated.

Control limits for $\bar{x}-$chart	Control limits for $R-$chart
$CL = \bar{\bar{x}}$	$CL = \bar{x}$
$UCL = \bar{\bar{x}} + A_2\bar{R}$	$UCL = D_4\bar{R}$
$LCL = \bar{\bar{x}} - A_2\bar{R}$	($LCL = D_3\bar{R}$ is not used.)

Coefficients of A_2/D_4

Group size, n	2	3	4	5	6
A_2	1.880	1.023	0.729	0.577	0.483
D_4	3.267	2.575	2.282	2.115	2.004

5) Graphs and a table are prepared based on the results of processing.

Data Input

(1) There are two methods of data input: one is to input data from the keyboard in response to a request for input, and the other is to input data from a cassette tape on which appropriate data has been previously stored by this program.
When the program is started, it asks which input method is to be used. Press the K key for input from the keyboard. Press the T key for input from cassette tape.

1) When input from the keyboard is selected
- Key in the group size n (number of data items in a group). The range of n is from 2 to 6.
- Key in the number of groups k. The range of k is from 2 to 40.
- Each time a group number is displayed, input the measured data associated with that group.

2) When input from cassette tape is selected
Connect the cassette tape recorder. Pressing the CR key (↵ key) starts the tape. The program then starts searching fro the appropriate data. When it finds the data, it displays the file name "$\bar{x}$–R DATA" and starts reading the data. First, the group size and the number of groups are read and displayed. Then the measured values are read.

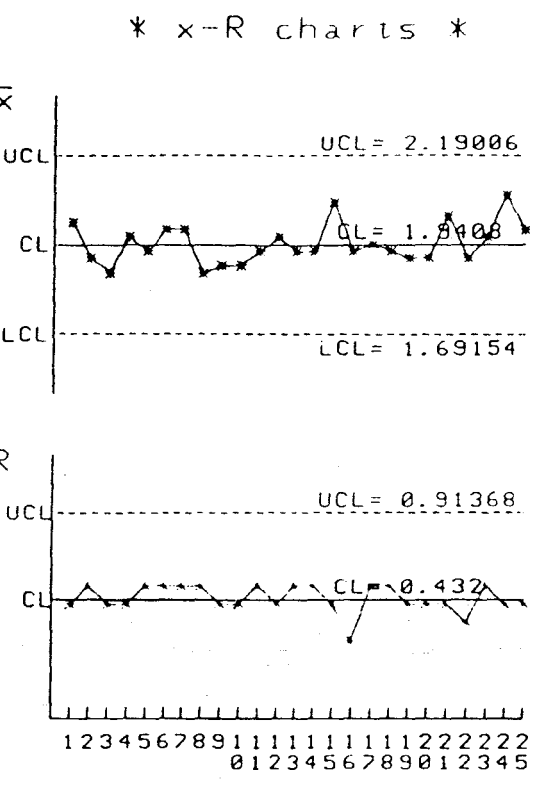
(2) After all the data is input, the program asks whether there are any measured values to be corrected. If there are, press the Y key.
If not, press the N key
When you press the Y key, the program prompts for the following input:
- Group number: The group containing the data to be corrected.
- Measured data number The data number in the specified group to be corrected.
- Input the correct value when the appropriate data is displayed.

The program then asks again whether there is any data to be corrected. If so, press the Y key. If not, press the N key.

Output of Result

(1) Printout of $\bar{x}$–R charts
Control charts (graphs) are plotted with the horizontal and vertical axes automatically set up according to the number of groups and the values of UCL and LCL. The groups within the control limits are represented by an asterisk (*), and the groups outside the control limits are represented by a Z symbol.
(2) Printout of table
The table contains the input data, the mean value $\bar{x}$ and the range R for each group.
(3) Saving input data (measured values) on cassette tape
The program asks whether or not the input data is to be saved on a cassette tape. If you wish to save the data on a cassette tape, press the Y key. If not, press the N key.
When you press the Y key, the program records the group size, the number of groups, and input data under the file name "$\bar{x}$–R DATA." (To start this save operation, set the cassette tape recorder, press the REC button, and press the ↵ key.)
After the input data is saved, program execution is terminated.
When you press the N key, the program is immediately terminated (without performing the save operation).

Graph and data output



Group number

Group number	x 1	x 2	x 3	x 4	x 5	mean x	R
1	2.01	1.81	2.21	1.81	2.21	2.001	0.401
2	1.71	1.71	2.21	1.81	2.11	1.901	0.501
3	1.71	2.11	1.91	1.71	1.91	1.861	0.401
4	1.91	2.01	1.91	2.21	1.81	1.961	0.401
5	1.81	1.81	1.71	2.21	2.11	1.921	0.501
6	2.11	2.21	2.11	1.71	1.81	1.981	0.501
7	2.21	1.81	2.01	1.71	2.21	1.981	0.501
8	2.21	1.71	2.01	1.71	1.71	1.861	0.501
9	2.11	1.71	1.91	2.01	1.71	1.881	0.401
10	1.71	2.11	1.71	2.11	1.81	1.901	0.401
11	2.21	2.21	1.81	1.71	1.71	1.921	0.501
12	2.21	1.81	2.21	1.81	1.81	1.961	0.401
13	2.01	2.01	1.71	1.71	2.21	1.921	0.501
14	2.21	1.91	1.91	1.71	1.91	1.921	0.501
15	1.81	2.21	2.21	2.01	2.11	2.061	0.401
16	1.91	2.01	1.91	1.81	2.01	1.921	0.201
17	2.21	1.91	1.71	1.81	2.11	1.941	0.501
18	2.01	1.71	2.21	1.91	1.81	1.921	0.501
19	2.01	2.11	1.71	1.81	1.91	1.901	0.401
20	2.01	1.71	1.81	2.11	1.91	1.981	0.401
21	1.91	2.21	2.21	1.81	2.01	2.021	0.401
22	1.81	1.71	2.01	2.01	2.01	1.981	0.301
23	1.71	2.21	1.71	2.01	2.21	1.961	0.501
24	2.21	2.21	2.21	2.01	1.81	2.081	0.401
25	2.21	2.01	2.11	1.81	1.81	1.981	0.401
					Total	48.921	10.401

LIST

4863 Bytes

```
  100 '
  110 CLEAR 1024:I=40:OPTION BASE 1:DIM A2(6),D4(6)
,X(I,6),XM(I),R(I)
  120 '
  130 RESTORE 150:FOR I=2 TO 6:READ A2(I):NEXT I
  140 FOR I=2 TO 6:READ D4(I):NEXT I
  150 DATA 1.88,1.023,0.729,0.577,0.483
  160 DATA 3.267,2.575,2.282,2.115,2.004
  200 '
  210 TI$="    * x-R charts *"
  220 CLS:PRINT TI$
```

```
  230 PRINT "Data input method":PRINT "  Keyboard":
PRINT "  Tape ":PRINT "< select K or T >"
  240 Z$=INKEY$:IF Z$="K" OR Z$="k" THEN GOSUB 1000
:GOTO 310
  250 IF Z$="T" OR Z$="t" THEN GOSUB 1100 ELSE GOTO
 240
  300 '
  310 CLS:PRINT TI$
  320 PRINT "Correcting data":LOCATE 8,2:PRINT"(Y o
r N):";
  330 Z$=INKEY$:IF Z$="Y" OR Z$="y" THEN PRINT CHR$
(14);Z$:GOTO 350
  340 IF Z$="N" OR Z$="n" THEN 400 ELSE 330
  350 INPUT "Group Number    :",GT:IF GT<1 OR GT>K T
HEN 350
  360 INPUT "Measured data number :",XT:IF XT<1 OR
XT>N THEN 360
  370 PRSNT X(GT,XT);:INPUT "-->",X(GT,XT):PRINT CH
R$(14):GOTO 320
  400 '
  410 CLS:PRINT TI$:PRINT "Being computed"
  420 XSUM=0:RSUM=0:LM=0:MM=X(1,1)
  430 FOR I=1 TO K:SX=0:MAX=X(I,1):MIN=MAX:PRINT "*
";
  440 FOR J=1 TO N:L=0:SX=SX+X(I,J)
  450 IF MAX<X(I,J) THEN MAX=X(I,J)
  460 IF MIN>X(I,J) THEN MIN=X(I,J)
  470 IF FRAC(X(I,J)*10^L)<>0 THEN L=L+1:GOTO 470
  480 IF LM<L THEN LM=L
  490 NEXT J
  500 XM(I)=SX/N:R(I)=MAX-MIN:IF MM<MAX THEN MM=MAX
  510 NEXT I:LM=-(LM+1):PRINT ">"
  520 FOR I=1 TO K:XM(I)=ROUND(XM(I),LM-1):XSUM=XSU
M+XM(I):R(I)=ROUND(R(I),LM-1):RSUM=RSUM+R(I):NEXT I
  530 RM=ROUND(RSUM/K,LM-3):RL=RM:RUCL=D4(N)*RM
  540 XM=ROUND(XSUM/K,LM-3):XCL=XM:XUCL=XM+A2(N)*RM
:XLCL=XM-A2(N)*RM
  600 '
  610 PRINT "Print out":GOSUB 1300
  620 LPRINT "O0,-10":LPRINT "L0":LPRINT "D10,0,10,
-50":LPRINT "D10,-25,90,-25"
  630 LPRINT "B2":LPRINT "L1":LPRINT "D90,-40,10,-4
0":LPRINT "D10,-10,90,-10"
```

```
  640 X=56:Y=-9:Z$="UCL="+LEFT$(STR$(XUCL),10):GOSU
B 1400:Y=-24:Z$=" CL="+LEFT$(STR$(XCL),10):GOSUB 14
00:Y=-43.4:Z$="LCL="+LEFT$(STR$(XLCL),10):GOSUB 140
0
  650 X=1.6:Y=-41.2:Z$="LCL":GOSUB 1400:Y=-26.2:Z$=
" CL":GOSUB 1400:Y=-11.2:Z$="UCL":GOSUB 1400
  660 X=0:Y=-2:Z$="x":LPRINT "S2":GOSUB 1400:Y=0:Z$
="-":GOSUB 1400:LPRINT "S1"
  670 LPRINT "J1":YS=(XUCL-XLCL)/30:ZC=XCL:ZU=XUCL:
ZL=XLCL:YL=-50:J=1:I=1:Z=XM(1):GOSUB 1500:LPRINT "M
";X;",";Y:GOSUB 1600
  680 FOR I=2 TO K:Z=XM(I):GOSUB 1500:LPRINT "D,";X
;",";Y:GOSUB 1600:NEXT I
  690 '
  700 LPRINT "J0":LPRINT "00,-70":LPRINT "L0":LPRIN
T "D10,0,10,-45,90,-45"
  710 FOR I=K TO 1 STEP -1:X=I*80/K+10:LPRINT "D";X
;",-43,";X;",-45":NEXT I
  720 X=10:Y=-57:Z$="Group number":GOSUB 1400
  730 FOR I=1 TO K:X=I*80/K+9.2:Y=-50:Z$=MID$(STR$(
I),2,1):GOSUB 1400:IF I>9 THEN Y=-53.6:Z$=RIGHT$(ST
R$(I),1):GOSUB 1400
  740 NEXT I
  750 LPRINT "D10,-25,90,-25":LPRINT "L1":LPRINT "D
90,-10,10,-10"
  760 X=0:Y=-3:Z$="R":LPRINT "S2":GOSUB 1400:LPRINT
 "S1":X=2.6:Y=-11.2:Z$="UCL":GOSUB 1400:Y=-26.2:Z$=
" CL":GOSUB 1400
  770 X=56:Y=-24:Z$=" CL="+LEFT$(STR$(RL),10):GOSUB
 1400:Y=-9:Z$="UCL="+LEFT$(STR$(RUCL),10):GOSUB 140
0
  780 YS=(RUCL-RL)/15:ZC=RL:ZU=RUCL:ZL=0:YL=-45:J=2
:I=1:Z=R(1):GOSUB 1500:LPRINT "M";X;",";Y:LPRINT "J
2":GOSUB 1600
  790 FOR I=2 TO K:Z=R(I):GOSUB 1500:LPRINT "D,";X;
",";Y:GOSUB 1600:NEXT I:LPRINT "J0"
  800 '
  810 X=LEN(STR$(INT(MM)))-1:FOM$="":IF LM<>0 THEN
FOM$=".":FOR I=1 TO -LM-1:FOM$=FOM$+"#":NEXT I
  820 FOR I=1 TO X:FOM$="#"+FOM$:NEXT I
  830 IF LEN(FOM$)>7 THEN FOM$="!"+LEFT$("#"+FOM$,8
):FOM1$=FOM$:GOTO 850
  840 FOR I=1 TO 8-LEN(FOM$):FOM$="#"+FOM$:NEXT I:F
OM1$="!"+RIGHT$(FOM$,7)+"#":FOM$="!"+FOM$
```

```
  850 GOSUB 1700:LPRINT CHR$(28);CHR$(46):LPRINT CH
R$(27);"S0"
  855 LPRINT L1$:LPRINT"| Group |          |          |
      |         |         |"
  860 LPRINT "|number";:FOR I=1 TO N:LPRINT "|  x̄";
I;"  ";:NEXT I:LPRINT "| mean x |    R    |"
  870 FOR I=1 TO K:LPRINT L2$:LPRINT USING "|######
";I;:FOR J=1 TO N:LPRINT USING FOM$;X(I,J);:NEXT J:
LPRINT USING FOM1$;XM(I);R(I);:LPRINT "|":NEXT I
  880 LPRINT L3$:LPRINT "|";TAB(N*9+2);"Total";:LPR
INT USING FOM1$;XSUM;RSUM;:LPRINT "|":LPRINT L4$:LP
RINT CHR$(27);"S1"
  890 FOR I=1 TO 4:LPRINT:NEXT I
  900 '
  910 CLS:PRINT TI$
  920 PRINT "Save data on tape ?";
  930 Z$=INKEY$:IF Z$="Y" OR Z$="y" THEN GOSUB 1200
:END
  940 IF Z$="N" OR Z$="n" THEN PRINT Z$:END ELSE 93
0
 1000 '
 1010 CLS:PRINT CHR$(14);TI$
 1020 INPUT "Group size(2~6):",N:IF N<2 OR N>6 THEN
 1020
 1030 PRINT"Number of group":INPUT "          (2~40)
:",K:IF N<2 OR N>6 THEN 1030
 1040 FOR I=1 TO K:PRINT:PRINT USING "Group number
  :#";I
 1050 FOR J=1 TO N:PRINT USING "Measred value##:";J
;:INPUT"",X(I,J):NEXT J
 1060 NEXT I:PRINT CHR$(14)
 1070 RETURN
 1100 '
 1110 CLS:PRINT TI$
 1120 PRINT "Input <x-RDATA> from tape. Connect the
 cassette tape recorder and press the CR key."
 1130 IF INKEY$<>CHR$(13) THEN 1130
 1140 OPEN "CAS0:x-RDATA" FOR INPUT AS #1
 1150 INPUT #1,N,K:PRINT "Group size :";N:PRINT "Nu
mber of group :";K
 1160 FOR I=1 TO K:FOR J=1 TO N:INPUT #1,X(I,J):NEX
T J,I
 1170 CLOSE #1
 1180 RETURN
 1200 '
```

```
 1210 CLS:PRINT TI$
 1220 PRINT "Save <x-RDATA> on tape. Connect the ca
ssette tape recorder and press the CR key.";
 1230 IF INKEY$<>CHR$(13) THEN 1230
 1240 OPEN "CAS0:x-RDATA" FOR OUTPUT AS #1
 1250 PRINT #1,N;",";K:FOR I=1 TO K:FOR J=1 TO N:PR
INT #1,X(I,J):NEXT J,I
 1260 CLOSE #1
 1270 RETURN
 1300 '
 1310 LPRINT CHR$(28);CHR$(37)
 1320 LPRINT "J0":LPRINT "Z3,1":LPRINT "S2":LPRINT
"Q0":LPRINT "Y0"
 1330 X=10:Y=0:Z$=TI$:GOSUB 1400:LPRINT "S1"
 1340 RETURN
 1400 '
 1410 LPRINT "M";X;",";Y:LPRINT "P";Z$:RETURN
 1500 '
 1510 X=I*80/K+10:Y=(Z-ZC)/YS-25:IF Z>ZU OR Z<ZL TH
EN P=8 ELSE P=3
 1520 LPRINT "L0":IF Y>0 THEN Y=0:LPRINT "L2"
 1530 IF Y<YL THEN Y=YL:LPRINT "L2"
 1540 RETURN
 1600 '
 1610 IF P=8 THEN LPRINT "J3":LPRINT "N8":LPRINT "J
1" ELSE LPRINT "N3"
 1620 RETURN
 1700 '
 1710 L10$=" ├──────":L11$="┼──────────":L12$="┴──────
──":L1$=" ┌──────":L2$=L10$:L3$=L10$:L4$=" └──────"
 1720 FOR I=1 TO N+2:L1$=L1$+"┬──────────":L2$=L2$+L1
1$:NEXT I:L1$=L1$+"┐":L2$=L2$+"│"
 1730 FOR I=1 TO N:L3$=L3$+L12$:L4$=L4$+"──────────"
:NEXT I:L3$=L3$+L11$+L11$+"│":L4$=L4$+L12$+L12$+"┘"
 1740 RETURN
```

DEBUG

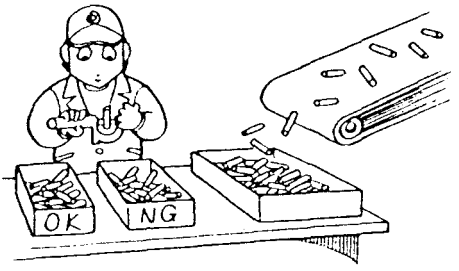
During data input, a 'MOUNT error' may occur when reading data from a cassette tape. In this case, execute 'MOUNT 1' in direct mode.
If an 'RW error' errors while saving data on a cassette tape, adjust the VOL dial on the cassette tape recorder (the TONE dial should be set to HIGH).

P-Pn CONTROL CHART

Program Outline

P– and Pn–charts are used to obtain data about defective parts at various manufacturing plants. The P-chart is used when the number n of pieces to be checked is variable. The Pn–chart is used when the number n of pieces to be checked is constant. The former is a control chart representing the percentage of defects, while the latter is a control chart representing the number of defects.

This program writes data to and reads data from CETL, and processes the data for output to a plotter-printer in the form of control charts.



Data Input

When the program is started, it asks whether or not the appropriate data is already stored in CETL. If the data is not yet stored in CETL, you should switch the MODE switch to CETL and input the data. Since CETL is described in detail in the CETL Reference Manual, data input to CETL is only briefly discussed here.

First, input the file name, the number of records, and the number of items using the N command. Input 6 as the number of items, and the number of data items + 1 as the number of records. Next, input data for each item as shown in List 1.

Use special care in inputting data under 'Expression.' The number 0 immediately after FL (see List 1) represents a file number. When a different file is specified, change 0 to the associated file number.

As a prompt for data input, the screen displays "1–1 ?". The first figure represents the record number and the second figure represents the item number. Since the data to be input comprises only the number of pieces to be inspected (item 2) and the number of defective pieces (item 3), you should only input the relevant data. For the prompts for irrelevant data, simply press the ↵ key. (See List 1.)

In the final record, input the previously calculated total number of inspected pieces (Σn) and the total number of defective pieces (ΣPn). This is the reason why the actual number of records + 1 should be input as the number of records. The number of groups should be 20 to 40.

To check that the data has been input correctly, use the T command to output the contents of data cells to the display unit or printer. If an error is found, use the J command for correction.

Groups are classified according to the lot number, etc. It is recommended that the grouping be done such that the number of pieces inspected is 50 to 100 per group and the number of defective pieces (Pn) per group is 3 to 4 on the average.

It should again be noted that the number of records to be input is the actual number of

records + 1.
After you input the data, reset the MODE switch to BASIC and start the program. Press any key on the keyboard and the program will ask whether a P—chart or a Pn—chart is required. Select the appropriate chart.
Next input the data file number and the number of groups. The plotter then plots the specified control chart.
This program (more precisely, CETL) performs the following calculations:

— Defect percentage of each group $P = Pn/n \times 100\ (\%)$
— Average defect percentage $\overline{P} = \Sigma Pn/\Sigma n \times 100\ (\%)$
— Control limits $CL = \overline{P}$

$$UCL = \overline{P} + 3\sqrt{\overline{P}(1-\overline{P})/n}$$

$$LCL = \overline{P} - 3\sqrt{\overline{P}(1-\overline{P})/n}$$

Data input list in CETL

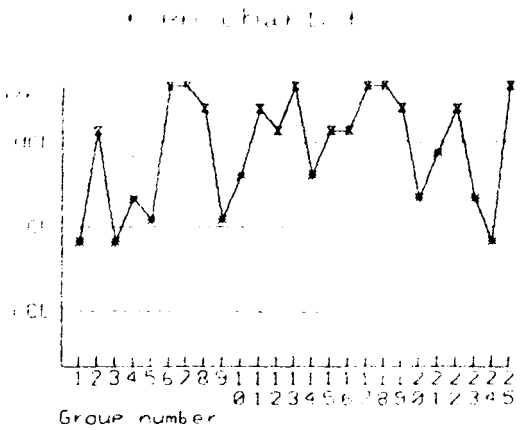
```
File Name:P-CHART

  Number of Rec.:26
  Number of Item:6


Label Record
Item1:GROUP
   Type(N/S)  :N
   Expression :RC
   Format     :##########

Item2:EXAMIN
   Type(N/S)  :N
   Expression ?
   Format     :##########

Item3:BAD
   Type(N/S)  :N
   Expression ?
   Format     :##########

Item4:BAD(%)
   Type(N/S)  :N
   Expression :IT(3)/IT(2)*100
   Format     :#####.####
```

```
Item5:UCL
   Type(N/S)   .N
   Expression :FL(0,26,4)+3*SQR(FL(0,26,
4)*(100-FL(0,26,4))/IT(2))
   Format      :######.###

Item6:LCL
   Type(N/S)   :N
   Expression :FL(0,26,4)-3*SQR(FL(0,26,
4)*100-FL(0,26,4)/IT(2))
   Format      :######.###

Data Area
1-1 ?
1-2 :50
1-3 :3
1-4 ?
1-5 ?
1-6 ?

2-1 ?
2-2 :50
2-3 :8
2-4 ?
2-5 ?
2-6 ?
```

■ Example of data for P-Pn chart

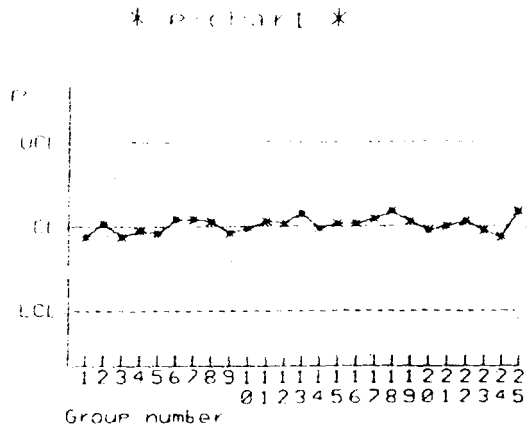
Group No.	Check n	Defects	Def.(%)				
1	50	3	6.0000	14	50	6	12.0000
2	50	8	16.0000	15	50	8	16.0000
3	50	3	6.0000	16	50	8	16.0000
4	50	5	10.0000	17	50	10	20.0000
5	50	4	8.0000	18	50	13	26.0000
6	50	10	20.0000	19	50	9	18.0000
7	50	10	20.0000	20	50	5	10.0000
8	50	9	18.0000	21	50	7	14.0000
9	50	4	8.0000	22	50	9	18.0000
10	50	6	12.0000	23	50	5	10.0000
11	50	9	18.0000	24	50	3	6.0000
12	50	8	16.0000	25	50	13	26.0000
13	50	12	24.0000	26	1250	187	14.9600

■ Example of printer output for Pn chart



CL= 2.4844%
UCL= 15.0463%
LCL= -0.0775%

■ Example of printer output for P chart



CL= 14.9600%

LIST

2024 Bytes

```
  100 '
  110 CLEAR 1024:DIM T$(1),P$(1)
  120 T$(0)="    * p-chart *":T$(1)="    * pn-chart *
":P$(0)="p":P$(1)="pn"
  200 '
  210 CLS:PRINT " *  p or pn chart  *"
  220 PRINT "Input data from CETL. If there is no d
ata in CETL,create data in CETL mode.       push key
 -> start";
  230 IF INKEY$="" THEN 230
  240 CLS:PRINT " *  p or pn chart  *"
  250 PRINT "[ P : p-chart  ]":PRINT "[ N : pn-char
t ]":PRINT "select P/N "
  260 Z$=INKEY$:IF Z$="P" OR Z$="p" THEN TI=0:GOTO
300
  270 IF Z$="N" OR Z$="n" THEN TI=1:GOTO 300 ELSE 2
60
  300 '
  310 CLS:PRINT T$(TI)
```

```
  320 INPUT "data file no.: ",F:IF F<0 OR F>9 THEN
320
  330 INPUT "Number of groups:",G:IF G<20 OR G>40 T
HEN 330
  400 '
  410 LPRINT CHR$(28);CHR$(37):LPRINT "J0":LPRINT "
Z2,6":LPRINT "Q0":LPRINT "Y0":LPRINT "B2"
  420 X=10:Y=0:Z$=T$(TI):LPRINT "S2":GOSUB 800:LPRI
NT "S1"
  430 LPRINT "O0,-10":LPRINT "L0":LPRINT "D10,0,10,
-50,90,-50"
  440 FOR I=G TO 1 STEP -1:X=I*80/G+10:LPRINT "D";X
;",-48,";X;",-50":NEXT I:X=10:Y=-60:Z$="Group numbe
r":GOSUB 800
  450 FOR I=1 TO G:X=I*80/G+9.2:Y=-53:Z$=MID$(STR$(
I),2,1):GOSUB 800:IF I>9 THEN Y=-56.6:Z$=RIGHT$(STR
$(I),1):GOSUB 800
  460 NEXT I
  470 LPRINT "L1":LPRINT "D90,-40,10,-40":LPRINT "L
0":LPRINT "D10,-25,90,-25":LPRINT "L1":LPRINT "D90,
-10,10,-10":LPRINT "L1"
  480 X=0:Y=-3:Z$=P$(TI):LPRINT "S2":GOSUB 800:LPRI
NT "S1"
  490 X=1.6:Y=-11.2:Z$="UCL":GOSUB 800:Y=-26.2:Z$="
 CL":GOSUB 800:Y=-41.2:Z$="LCL":GOSUB 800:LPRINT "J
1"
  500 '
  510 ON TI+1 GOSUB 600,700
  520 LPRINT "H5":LPRINT "J0":LPRINT CHR$(28);CHR$(
46)
  530 ON TI+1 GOSUB 650,750
  540 LPRINT CHR$(27);"F5"
  550 GOTO 240
  600 '
  610 CL=FL(F,G+1,4):FOR I=1 TO G:UCL=FL(F,I,5):LCL
=FL(F,I,6):Z=FL(F,I,4):GOSUB 900
  620 IF I=1 THEN LPRINT "M";X;",";Y ELSE LPRINT "D
,";X;",";Y
  630 GOSUB 1000:NEXT I
  640 RETURN
  650 LPRINT USING " CL=###.####%";CL:RETURN
  700 '
  710 N=FL(F,1,2):PB=FL(F,G+1,4)/100:NP=N*PB:S1=3*S
QR(NP*(1-PB))
  720 CL=NP:UCL=NP+S1:LCL=NP-S1
```

```
  730 FOR I=1 TO G:Z=FL(F,I,4):GOSUB 900:IF I=1 THE
N LPRINT "M";X;",";Y ELSE LPRINT "D,";X;",";Y
  740 GOSUB 1000:NEXT I:RETURN
  750 '
  760 GOSUB 650:LPRINT USING "UCL=###.####%";UCL:LP
RINT USING "LCL=###.####%";LCL
  770 RETURN
  800 '
  810 LPRINT "M";X;",";Y:LPRINT "P";Z$:RETURN
  900 '
  910 X=I*80/G+10:Y=(Z-CL)/(UCL-LCL)*30-25:IF Z>UCL
 OR Z<LCL THEN P=8 ELSE P=3
  920 LPRINT "L0":IF Y>0 THEN Y=0:LPRINT "L2"
  930 IF Y<-50 THEN Y=-50:LPRINT "L2"
  940 RETURN
 1000 '
 1010 IF P=8 THEN LPRINT "J3":LPRINT "N8":LPRINT "J
1" ELSE LPRINT "N3"
 1020 RETURN
```

DEBUG

Although this program is not very complicated, it requires some understanding of the CETL commands becuase it is linked to CETL.
The CETL commands used in this program are N (create a new file), J (modify and input data), and T (display or print data in a file in the form of a table). For example, to output a list of input data on a printer, the T command is used as follows:

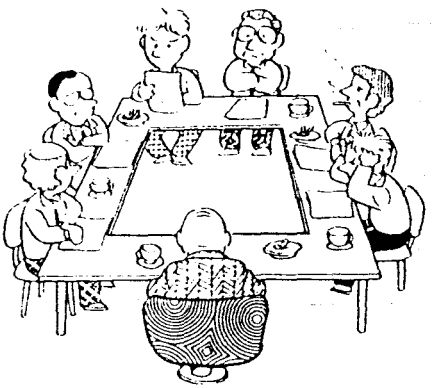
```
>T
Rec. ?, ↵
Item ? 1, 3 ↵
Printer (Y/N) ? Y ↵
```

To output all the data, change 'ITEM ? 1, 3 ↵ ' to 'ITEM ?, ↵ .'

BUDGET CONTROL

This application requires a floppy disk.

Program Outline



This program permits establishing a budget for each account title and controlling the appropriations from the budget by inputting the date and amount of an appropriation each time it is made. The items to be entered are the account titles, amount of the budget, and dates and amounts of appropriations. Each time these items are input, the program calculates the balance of the budget and the rate of execution of the budget. The capabilities of this program include screen display and storing of data, deletion of unnecessary account titles, and printout of stored data, as well as the totaling capability.

A floppy disk is used for a random access file for storing data, allowing data transfer at relatively high speeds.

Directions

This program uses two random access files. Therefore, you should execute MOUNT 2 in direct mode before executing the program. (Once this command has been executed, it is effective even after the power is turned off.) If a new floppy disk is used for storing the data, execute the FORMAT command.

When the program is started, it displays the following menu.

❶

```
** Budget control **

  [1] Store
  [2] Display
  [3] Delete
  [4] Print
  [5] End
```

❷

```
*** STORING ***

[1]Account title
[2]Budget execution
[3]Menu
```

By pressing any key from 1 to 5, it is possible to execute the associated operation. Note, however, that when this program is executed for the first time (when no data has been stored), operations 3 to 5 cannot be performed.

(1) Storing

When this mode is specified, the program displays the following menu ❷ :

1) Title of account

Pressing the 1 key permits storing an account. An account title should not exceed 13 characters in length, and the amount of the budget must not exceed 8 digits in length. When you input an account title and the amount of the budget, the program displays the prompt "OK? (Y/N)." Press anything except the N key if the input data is correct. If the input data is incorrect, press the N key and make the appropriate correction.

The program then asks for the date and amount of the first budget item. When you input the date and the amount (max. 8 digits), the program displays the balance of the budget (i.e., the amount of the original budget minus the amount of the appropriation) and the current rate of buget execution. It then displays the prompt "OK? (Y/N)." Press anything except the N key if the data is correct. Press the N key if the data is incorrect, and make the appropriate correction. Now, the program returns to the Catalog menu.

2) Budget execution

In this mode, the file in which account titles have been stored can be updated. The program displays the first account title and associated budget in this file. Retrieve the desired account title by pressing the cursor keys ← and → . When it is found, press the Y key and input the date and the amount of appropriation in the same manner as the input of budget execution for storing an account title. The program then calculates the current budget balance and the rate of budget execution. They are stored when you press the Y key. The program then returns to the menu.

3) Menu

The program moves from the current Catalog menu to the Budgetary Control menu.

2) Display

ressing the 2 key puts the program into display mode. In this mode, the program dis-lays the data initially stored in the file. An example of the display is shown below.

❸

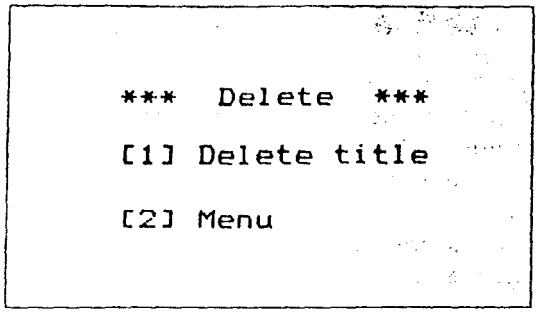
```
Title  AD. expenses
Budget      $    1000
====================
Date          01/09/83
Amt.of B.A $     200
Balance    $     800
Rate of B.E 20.000%
 Move[CSR] Menu[E]
```

ressing the cursor keys ← and → causes the account title and budgets to be display-d item by item by scrolling. To display the budget execution, use the cursor keys ↓ nd ↑ . To return to the menu, press the E key.

) Delete

ressing the 3 key causes the program to enter delete mode and display the following enu:

❹



1) Delete account title

By pressing the 1 key, it is possible to delete unnecessary title-of-account files. In this mode, the program displays account titles and budgets from the beginning. Using the cursor keys ← and → , scroll the screen display until you find the title-of- account file to be deleted. When that file is displayed, press the Y key and the file will be deleted. When you press the N key, the program returns to the menu without making any deletion.

2) Menu

The program moves from the current delete menu to the Budget Control menu.

(4) Printout

In this mode, it is possible to output all the stored data, together with the totals, on a printer. An example of the printout is shown below.

The "total" fields included in the output indicate the cumulative totals of amounts of budget execution, the current budget balance, and the cumulative totals of rates of budget execution (from left to right).

(5) End

Terminates program execution.

❺

[Account title] AD. expenses [Budget] $ 1,000

[Date]	[Amount of B.A.]	[Balance]	[Rate of BE]
01/09/83	$ 200	$ 800	20.000%
10/10/83	$ 100	$ 700	10.000%
Total	$ 300	$ 700	30.000%

LIST

LIST 1 *84 Bytes*
LIST 2 *5182 Bytes*

LIST 1

```
  10 INPUT "FILE NAME";N$
  20 OPEN N$ FOR OUTPUT AS #1
  30 FOR A=1 TO 10
  40 PRINT #1,"*":NEXT
  50 CLOSE
```

Only necessary for creating file for the first-time

LIST 2

```
 100 ' Budget control
 110 '
 120 FIELD#1,3 AS A1$,13 AS B1$,8 AS C1$,8 AS D1$,
 AS E1$
 130 FIELD#2,3 AS F1$,1 AS G1$,8 AS H1$,8 AS I1$,8
AS J1$,8 AS K1$
 140 GOTO 760
 150 '
 160 CLS:GOSUB 1630
 170 OPEN "MAIN" AS #1
 180 ME=LOF(1)
 190 OPEN "DATA" AS #2
 200 DE=LOF(2)
 210 GOSUB 1650
 220 P=1
 230 GET #1,P
 240 IF E1$="*" THEN P=P+1:IF P<ME THEN 230 ELSE 7
0
 250 GOSUB 520:LOCATE 0,7:PRINT "Move[Cursor]
  Menu[E]";
 260 Q=1
 270 GET #2,Q
 280 IF VAL(F1$)<>VAL(A1$) OR G1$="*" THEN Q=Q+1:I
 Q<DE THEN 270 ELSE 760
 290 GOSUB 570
 300 GOSUB 1620
 310 Z$=INKEY$:IF Z$="" THEN 310
 320 IF Z$="E" THEN CLOSE :GOTO 760
 330 GOSUB 1630
 340 IF Z$=CHR$(28) THEN P=P+1:T=0:GOTO 390
 350 IF Z$=CHR$(29) THEN P=P-1:T=1:GOTO 440
 360 IF Z$=CHR$(30) THEN Q=Q-1:GOTO 460
 370 IF Z$=CHR$(31) THEN Q=Q+1:GOTO 490
```

```
  380 GOTO 300
  390 IF P>ME THEN P=1
  400 GET #1,P
  410 IF E1$="*" AND T=0 THEN P=P+1:GOTO 390
  420 IF E1$="*" AND T=1 THEN P=P-1:GOTO 440
  430 GOSUB 540:Q=1:GOTO 500
  440 IF P<1 THEN P=ME
  450 GOTO 400
  460 IF Q<1 THEN Q=DE
  470 GET #2,Q:IF VAL(F1$)<>VAL(A1$) OR G1$="*" THE
N Q=Q-1:GOTO 460
  480 GOSUB 570:GOTO 300
  490 IF Q>DE THEN Q=1
  500 GET #2,Q:IF VAL(F1$)<>VAL(A1$) OR G1$="*" THE
N Q=Q+1:GOTO 480
  510 GOTO 480
  520 '
  530 CLS
  540 LOCATE 0,0:PRINT "Title of account";TAB(6);B1
$
  550 LOCATE 0,1:PRINT "Budget";TAB(10);"$";I1$
  560 RETURN
  570 LOCATE 0,3:PRINT "Date";TAB(11);H1$
  580 LOCATE 0,4:PRINT "Budget excecution";TAB(10);
"$";I1$
  590 LOCATE 0,5:PRINT "Balance";TAB(10);"$";J1$
  600 LOCATE 0,6:PRINT "Rate of budget excecution";
TAB(11);K1$
  610 RETURN
  620 '
  630 OPEN "MAIN" AS #1
  640 ME=LOF(1):P=ME+1:IF P<1 THEN P=1
  650 A$=STR$(P)
  660 INPUT "Title of account";B$
  670 INPUT "Budget";C$
  680 D$=C$
  690 E$=" "
  700 GOSUB 1560
  710 PUT #1,P
  720 P=P+1
  730 Z$=INKEY$:IF Z$="" THEN 730
  740 IF Z$="E" THEN CLOSE #1:END
  750 GOTO 650
  760 CLS:CLOSE:PRINT "** Budget control **";
  770 PRINT
```

```
 780 PRINT TAB(2);"[1] Store"
 790 PRINT TAB(2);"[2] Display"
 800 PRINT TAB(2);"[3] Delete"
 810 PRINT TAB(2);"[4] Print"
 820 PRINT TAB(2);"[5] End"
 830 Z$=INKEY$:IF Z$="" THEN 830
 840 Z=VAL(Z$):IF Z<1 OR Z>5 THEN 830
 850 ON Z GOTO 870,150,2040,1700,860
 860 CLS:END
 870 '
 880 CLS:PRINT "*** STORING ***"
 890 PRINT
 900 PRINT "[1]Title of account"
 910 PRINT "[2]Budget excecution";
 920 PRINT "[3]Menu"
 930 Z$=INKEY$:IF Z$="" THEN 930
 940 Z=VAL(Z$):IF Z<1 OR Z>5 THEN 930
 950 ON Z GOTO 960,1120,760
 960 CLS
 970 GOSUB 1630
 980 OPEN "MAIN" AS #1
 990 ME=LOF(1):P=ME+1:IF P<1 THEN P=1
1000 A$=STR$(P)
1010 CLS:PRINT "---STORING ---"
1020 INPUT "Title of account";B$
1030 PRINT "Budget";TAB(10);:INPUT C$
1040 D$=C$
1050 LOCATE 1,7:PRINT "**** OK?[Y/N] ****";
1060 Z$=INKEY$:IF Z$="" THEN 1060
1070 IF Z$="N" THEN CLS:GOTO 1010
1080 GOSUB 1640
1090 GOSUB 1560
1100 PUT #1,P
1110 GOTO 1310
1120 CLS
1130 GOSUB 1630
1140 OPEN "MAIN" AS #1
1150 ME=LOF(1):IF ME<1 THEN 1680
1160 CLS
1170 LOCATE 0,3
1180 PRINT "Select title [<=][=>]"
1190 PRINT "Renewal file [Y]"
1200 PRINT "Cancel  [N]"
1210 P=1:GOTO1260
1220 GOSUB 1620
```

```
 1230 S=0:GOSUB 1480
 1240 IF P<1 THEN P=ME
 1250 IF P>ME THEN P=1
 1260 GET #1,P
 1270 IF E1$="*" AND T=0 THEN P=P+1:GOTO 1240
 1280 IF E1$="*" AND T=1 THEN P=P-1:GOTO 1240
 1290 GOSUB 540
 1300 GOTO 1220
 1310 OPEN "DATA" AS #2
 1320 DE=LOF(2):Q=DE+1
 1330 F$=A1$
 1340 G$=" "
 1350 LOCATE 0,3:PRINT "[Date]"TAB(10);:INPUT H$
 1360 PRINT "[Budget excecution]";TAB(10);:INPUT I$
 1370 J#=(VAL(D1$)-VAL(I$)):J$=STR$(J#)
 1380 PRINT "[Blance]";TAB(10);J$
 1390 K$=LEFT$(STR$(ROUND(VAL(I$)/VAL(C1$),-6)*100)
,7)+"%"
 1400 PRINT "[Budget excecution ]";TAB(10);K$
 1410 PRINT " ** OK?[Y/N] **";
 1420 Z$=INKEY$:IF Z$="" THEN 1420
 1430 GOSUB 1640
 1440 IF Z$="N" THEN GOSUB 530:GOTO 1330
 1450 RSET D1$=J$:PUT #1,P
 1460 GOSUB 1960:PUT #2,Q
 1470 CLOSE :GOTO 870
 1480 Z$=INKEY$:IF Z$="" THEN 1480
 1490 GOSUB 1630
 1500 IF Z$="Y" AND S=0 THEN GOSUB 530:GOTO 1310
 1510 IF Z$="Y" AND S=1 THEN 2310
 1520 IF Z$="N" THEN 760
 1530 IF Z$=CHR$(28) THEN P=P+1:T=0:IF P>ME THEN P=
ME
 1550 RETURN
 1560 RSET A1$=A$
 1570 RSET B1$=B$
 1580 RSET C1$=C$
 1590 RSET D1$=D$
 1600 RSET E1$=E$
 1610 RETURN
 1620 LOCATE 0,2:PRINT "====================":RETURN
 1630 LOCATE 6,2:PRINT "reading":RETURN
 1640 LOCATE 0,7:PRINT "                    ";:RETUR
N
 1650 IF ME=>1 OR DE=>1 THEN RETURN
```

```
 1660 CLS:LOCATE 4,2:PRINT "File not stored"
 1670 FOR Z=0 TO 300:NEXT Z:GOTO 760
 1680 CLS:LOCATE 5,2:PRINT "Title not stored"
 1690 FOR Z=0 TO 300:NEXT Z:CLOSE:GOTO 870
 1700 '
 1710 CLS:GOSUB 1630
 1720 OPEN "MAIN" AS #1
 1730 OPEN "DATA" AS #2
 1740 ME=LOF(1):DE=LOF(2)
 1750 GOSUB 1650
 1760 LOCATE 4,2:PRINT "now printing"
 1770 FOR P=1 TO ME
 1780 I=0:K=0
 1790 GET #1,P:IF E1$="*" THEN 1940
 1800 LPRINT "[Title of account]";TAB(8);B1$;TAB(30
);"[Budget]";TAB(42);
 1810 LPRINT USING "¥###,###,###";VAL(C1$)
 1820 LPRINT:LPRINT "[Date]";TAB(15);"[Budget excec
ution"]";
 1830 LPRINT TAB(32);"[Balance]";TAB(45);"[Rate of
budget execution]"
 1840 FOR Q=1 TO DE
 1850 GET #2,Q
 1860 IF VAL(F1$)<>VAL(A1$) OR G1$="*" THEN 1900
 1870 LPRINT H1$;TAB(13);:LPRINT USING "¥###,###,###
";VAL(I1$);
 1880 LPRINT USING "      ¥###,###,###      ###.###%";
VAL(J1$),VAL(LEFT$(K1$,7))
 1890 I=I+VAL(I1$):K=K+VAL(LEFT$(K1$,7))
 1900 NEXT Q
 1910 FOR Z=0 TO 51:LPRINT "-";:NEXT Z:LPRINT "-"
 1920 LPRINT USING "Total          ¥###,###,###      ¥#
#,###,###      ###.###%";I,VAL(D1$),K
 1930 LPRINT:LPRINT:LPRINT
 1940 NEXT P
 1950 GOTO 760
 1960 '
 1970 RSET F1$=F$
 1980 RSET G1$=G$
 1990 RSET H1$=H$
 2000 RSET I1$=I$
 2010 RSET J1$=J$
 2020 RSET K1$=K$
 2030 RETURN
 2040 '
```

```
2050 CLS:PRINT " ***  Delete  ***"
2060 PRINT
2070 PRINT TAB(2);"[1] Delete title"
2080 PRINT
2090 PRINT TAB(2);"[2] Menu"
2100 Z$=INKEY$:IF Z$="" THEN 2100
2110 Z=VAL(Z$):IF Z<1 OR Z>2 THEN 2100
2120 ON Z GOTO 2130,760
2130 CLS
2140 GOSUB 1630
2150 OPEN "MAIN" AS #1
2160 ME=LOF(1):IF ME<1 THEN 1660
2170 CLS:LOCATE 0,3
2180 PRINT "Select title [<-]-[->]"
2190 PRINT "Delete title [Y]"
2200 PRINT "Cancel       [N]"
2210 P=1:GOTO 2260
2220 GOSUB 1620
2230 S=1:GOSUB 1480
2240 IF P<1 THEN P=ME
2250 IF P>ME THEN P=1
2260 GET #1,P
2270 IF E1$="*" AND T=0 THEN P=P+1:GOTO 2240
2280 IF E1$="*" AND T=1 THEN P=P-1:GOTO 2240
2290 GOSUB 540
2300 GOTO 2220
2310 RSET E1$="*":PUT #1,P:CLOSE #1:GOTO 2040
```

DEBUG

An SN error during program execution indicates an error in the line currently displayed. However, any other error which occurs during program execution may indicate an error in a line other than the line currently displayed.
If an error occurs after the menu is output, it is advisable to check the line (routine) to which a jump was made from the ON . . . GOTO statement following the routine to display the menu. Lines 850 and 930 are examples of lines which are to be checked first.
The routine for printout is included in lines 1800–1940.

Utility Programs

PAYMENTS PLAN FOR LONG-TERM LOANS

Program Outline

Long-term loans are a convenient means to purchase a home, new car, or other expensive articles. There are four modes of installment payment of a loan. They are equal installments of principal, equal installments of principal and interests, add-on, and installments of decreasing principal and interests.

These calculations can be performed by a desktop calculator capable of handling functions. This program permits performing the same calculations on the FP-200 with greater ease.

Directions

(1) Equal installments of principal

In this mode, the amount of monthly payment for the principal remains the same throughout the term of loan. Since the balance of loan decreases linearly, the amount of interest payment and the amount of principal plus interest decrease accordingly. Thus, the calculation is easy. However, this mode of payment is not realistic because the amount of payment in the early periods is relatively greater than that in the latter periods.

(2) Equal installments of principal plus interests

In this mode, the amount of interest for the entire term of loan for the total amount of loan is calculated without regard to the fact that the balance of loan long (e.g., housing loan), it is easy to make a payment plan.

(3) Add-on

In this mode, the amount of interest for the entire term of loan for the total amount of loan is calculated without regard to the fact that the balance of loan decreases gradually. Although the method of calculation is extremely easy, the net interest becomes much higher than other mode of payment.

The add-on system is often used for commodities sold on a short-term loan basis. This system should be avoided if you are a borrower.

(4) Installments of decreasing principal plus interests

In this mode, the monthly payment of principal and interest components is increased at a fixed rate when a chronic inflation or continual increase in income is expected. Since the calculation is complex, this program is particularly useful when using this mode of payment.

Data Input

Start the program, and it displays the menu asking the mode of payment desired. Select the mode and input the amount of loan, number of installments, and annual interest rate. The program then outputs the monthly payment of principal and interest components, together with the balance of loan.

LIST *3314 Bytes*

```
    10 ' Payment Plan for long-term Lorn
    20 FOR I=1 TO 4
    30 READ Z$(I)
    40 NEXT I
    50 LPRINT CHR$(27);"2"
    60 CLS
    70 '
    80 PRINT"1:Equal Inst. of        Principal"
    90 PRINT"2:Equal Inst. of        Principal Plus
     Interests"
   100 PRINT"3:Add-On"
   110 PRINT"4:Inst.of decreasing   Principal Plus";
:FOR I=0 TO 800:NEXT I:PRINT:PRINT"    Interests"
   120 PRINT"[Select 1~4 key]";
   130 X$=INKEY$:IF X$="" THEN 130
   140 IF ASC(X$)<ASC("1") THEN 130 ELSE IF ASC("4")
<ASC(X$) THEN 130
   150 X=VAL(X$)
   160 GOSUB 1000
   170 ON X GOSUB 2000,3000,4000,5000
   180 LPRINT CHR$(27);"2"
   190 CLS:PRINT"********************** If you wish t
o    **    continue,        ** PUSH <CONT> key. *******
***************"
   200 LOCATE 0,5
   210 STOP:GOTO 50
  1000 '
  1010 CLS
  1020 PRINT Z$(X)
  1030 PRINT:INPUT"Amount of Lorn                  =",A
#
  1040 PRINT:INPUT"Number of Inst.                 =",T
#
```

```
 1050 PRINT:INPUT"Annual interest rate          =",R
#
 1060 IF X=4 THEN PRINT:INPUT"Rate of increase
        =",S#
 1070 CLS
 1080 LPRINT"Payment Plan for long-term Lorn <<";Z$
(X);">>":LPRINT
 1090 LPRINT "Amount of lorn       : ";:LPRINT USIN
G"$##########";A#;
 1100 LPRINT TAB(40) "Number of inst.  :";:LPRINT U
SING"###";T#;:LPRINT"times"
 1110 LPRINT "Annual interset rate :";STR$(R#);"%";
 1120 IF X=4 THEN LPRINT TAB(40);"Rate of increase
:";STR$(S#);"%" ELSE LPRINT
 1130 LPRINT CHR$(27);"1"
 1140 LPRINT:LPRINT"┌──────┬──────────┬──────────┬──
──────┬──────────┐"
 1150 LPRINT"│No. │Principal │ Interest │  Total   │ Ba
lance  │"
 1160 RETURN
 2000 '
 2010 L#=A#
 2020 J#=A#/T#
 2030 FOR I=1 TO CSNG(T#)
 2040 K#=J#*(T#-CDBL(I)+1)*R#/1200
 2050 L#=L#-J#
 2060 IF L#<.5 THEN L#=0
 2070 LPRINT"├──────┼──────────┼──────────┼──────────┼──
──────┤"
 2080 LPRINT"│";:LPRINT USING"###";I;:LPRINT"│";
 2090 LPRINT USING"##########";J#;:LPRINT"│";
 2100 LPRINT USING"##########";K#;:LPRINT"│";
 2110 LPRINT USING"##########";J#+K#;:LPRINT"│";
 2120 LPRINT USING"##########";L#;:LPRINT"│"
 2130 NEXT I
 2140 LPRINT"└──────┴──────────┴──────────┴──────────┴──
──────┘"
 2150 RETURN
 3000 '
 3010 R#=R#/1200
 3020 L#=A#
 3030 C#=A#*R#/(1-(1+R#)^(-T#))
 3040 FOR I=1 TO CSNG(T#)
 3050 B#=C#/(1+R#)^(T#+1-CDBL(I))
 3060 L#=L#-B#
```

```
3070 IF L#<.5 THEN L#=0
3080 LPRINT"├──┼──────────┼──────────┼──────────┼──────────┤"
3090 LPRINT"│";:LPRINT USING"###";I;:LPRINT"│";
3100 LPRINT USING"##########";B#;:LPRINT"│";
3110 LPRINT USING"##########";C#-B#;:LPRINT"│";
3120 LPRINT USING"##########";C#;:LPRINT"│";
3130 LPRINT USING"##########";L#;:LPRINT"│"
3140 NEXT I
3150 LPRINT"└──┴──────────┴──────────┴──────────┴──────────┘"
3160 RETURN
4000 '
4010 L#=A#
4020 J#=A#/T#
4030 K#=(A#*R#/100*T#/12)/T#
4040 FOR I=1 TO CSNG(T#)
4050 L#=L#-J#
4060 IF L#<.5 THEN L#=0
4070 LPRINT"├──┼──────────┼──────────┼──────────┼──────────┤"
4080 LPRINT"│";:LPRINT USING"###";I;:LPRINT"│";
4090 LPRINT USING"##########";J#;:LPRINT"│";
4100 LPRINT USING"##########";K#;:LPRINT"│";
4110 LPRINT USING"##########";J#+K#;:LPRINT"│";
4120 LPRINT USING"##########";L#;:LPRINT"│"
4130 NEXT I
4140 LPRINT"└──┴──────────┴──────────┴──────────┴──────────┘"
4150 RETURN
5000 '
5010 R#=R#/1200
5020 S#=S#/1200
5030 M#=A#*(R#-S#)/(1-((1+S#)/(1+R#))^T#)
5040 K#=A#*R#
5050 L#=A#-M#+K#
5060 LPRINT"├──┼──────────┼──────────┼──────────┼──────────┤"
5070 LPRINT"│  1 │";
5080 LPRINT USING"##########";M#-K#;:LPRINT"│";
5090 LPRINT USING"##########";K#;:LPRINT"│";
5100 LPRINT USING"##########";M#;:LPRINT"│";
5110 LPRINT USING"##########";L#;:LPRINT"│"
5120 FOR I=2 TO CSNG(T#)
5130 N#=M#*(1+S#)^(CDBL(I)-1)
```

```
 5140 K#=L#*R#
 5150 L#=L#-N#+K#
 5160 IF L#<.5 THEN L#=0
 5170 LPRINT"├────┼──────────┼──────────┼──────────┼──────────┤"
 5180 LPRINT"│";:LPRINT USING"###";I;:LPRINT"│";
 5190 LPRINT USING"##########";N#-K#;:LPRINT"│";
 5200 LPRINT USING"##########";K#;:LPRINT"│";
 5210 LPRINT USING"##########";N#;:LPRINT"│";
 5220 LPRINT USING"##########";L#;:LPRINT"│"
 5230 NEXT I
 5240 LPRINT"└────┴──────────┴──────────┴──────────┴──────────┘"
 5250 RETURN
 5260 DATA Equal Inst. of Principal,Equal Inst. of
Principal Plus Interests,Add-On,Inst. of decreasing
 Principal Plus Interests
```

ACCOUNT BOOK
(HOUSEHOLD ACCOUNTS)

This application requires a floppy disk.

rogram Outline

his program records receipts and expenses and lculates the balance, sub-totals, and grand tal. It also permits making corrections easily. us, the program can be used for accounts anagement in small businesses and homes.

ata Input

fore loading this program, perform the following:
MOUNT 1 ↵
Execute the following 4-line program:

```
10  OPEN "A" FOR OUTPUT AS #1
20  FOR I = 0 TO 20: PRINT #1, "*"
30  NEXT
40  END
```

en, start this program. When the floppy disk is ready, press Y and ↵ for the mpt "DISK OK?". For the prompt "FILE NAME?", press A and ↵.
e menu is then displayed.
enu
Write
Edit
Sub-total
Total
Kill
Save
Vrite
st, enter anything other than Y for the prompt "END?". Then input the following a in this order:
ar, month, day
de (99)
mo
pense
ceipt
ssing anything other than Y when the prompt "END?" appears again causes a return he menu, allowing input of more data.
Edit
editing procedure is as follows:

M key Increment No.
I key Decrement No.
K key Year, month, day → Code → Memo → Expense → Receipt → Total.
J key Reverse the sequence of the K key.
E key Press this key before inputting new data.
N key Return to the menu.

3. Sub-total

First, input the year, month, day and the memo. Then, input one by one by numbers for the prompt "FROM, THRU". At the end of the input, the menu is displayed again.

4. Total

The total of the data that has been input is calculated. At the end of the processing, the menu is displayed again.

5. Kill

Input the number of the data which is to be killed, and the associated contents are displayed. The prompt "DATA KILL?" is displayed again for confirmation. Enter Y to kill the data. If you do not wish to kill the data, enter anything other than Y. After that, the menu is displayed again.

6. Save

First, the message "DISK OK?" is displayed. Enter Y when the disk is ready. For the prompt "FILE NAME?", enter A. After that, the menu is displayed again.

LIST *2102 Bytes*

```
    10 '
    20 '
    30 '
    40 CLS:PRINT "ACCOUNT BOOK"
    50 INPUT "DISK OK";A$:CLEAR:FILES:INPUT "FILE NA
ME ";N$:OPEN N$ FOR INPUT AS #1
    60 DIM D$(100,6)
    70 FOR I=0 TO 6:READ D$(0,I):NEXT
    80 DATA NO.         ,Year.month.day,Code        ,Mem
o        ,Expense    ,Receipt    ,Total
    90 A=A+1
   100 FOR I=0 TO 6:INPUT #1,D$(A,I):NEXT:IF D$(A,0)
="*" THEN A=A-1 ELSE GOTO 90
   110 CLOSE:PRINT "Number of data":PRINTTAB(3);A;"A
items":FOR I=0 TO 200:NEXT
   120 CLS:PRINT "MENU"
   130 PRINT "1)))Write":PRINT "2)))EDIT":PRINT "3))
)Sub-total":PRINT "4)))Total":PRINT "5)))KILL":PRIN
T "6)))SAVE "
```

```
  140 INPUT "Which?";B
  150 ON B GOTO 170,200,320,360,410,460
  160 GOTO 120
  170 CLS:PRINT "Write":PRINT "NEW DATA "
  180 A=A+1:PRINT "NO.";A:INPUT "END (Y)";Y$:IF Y$=
"Y" OR Y$="y" THEN A=A-1:GOTO 120 ELSE D$(A,0)=STR$
(A)
  190 FOR I=1 TO 5:PRINT D$(0,I):INPUT D$(A,I):NEXT
:D$(A,6)=STR$(VAL(D$(A-1,6))+VAL(D$(A,5))-VAL(D$(A,
4))):GOTO 180
  200 CLS:PRINT "EDIT"
  210 X=1:Y=1
  220 LOCATE 0,0:PRINT D$(0,X);"     "
  230 PRINT D$(Y,X);"     ":LOCATE 0,5:PRINT "NO.";
Y;"   "
  240 K$=INKEY$:IF K$="" THEN 240
  250 IF K$="I" THEN Y=Y-1:IF Y<1 THEN Y=A
  260 IF K$="M" THEN Y=Y+1:IF Y>A THEN Y=1
  270 IF K$="J" THEN X=X-1:IF X<0 THEN X=6
  280 IF K$="K" THEN X=X+1:IF X>6 THEN X=0
  290 IF K$="E" THEN INPUT "NEW DATA";D$(Y,X):D$(Y,
6)=STR$(VAL(D$(Y-1,6))+VAL(D$(Y,5))-VAL(D$(Y,4))):C
LS:GOTO 220
  300 IF K$="N" THEN 120
  310 GOTO 220
  320 CLS:PRINT "Sub-total DATA MAKE "
  330 PRINT "NO.";A+1:A=A+1:D$(A,0)=STR$(A):D$(A,2)
="99"
  340 FOR I=1 TO 3 STEP 2:PRINT D$(0,I):INPUT D$(A,
I):NEXT:PRINT "From,Thru(NO.)":INPUT E,F
  350 S#=0:W#=0:FOR I=E TO F:S#=S#+VAL(D$(I,4)):W#=
W#+VAL(D$(I,5)):NEXT:D$(A,4)=STR$(S#):D$(A,5)=STR$(
W#):D$(A,6)=D$(A-1,6):GOTO 120
  360 CLS:PRINT "Total "
  370 A=A+1:D$(A,0)=STR$(A)
  380 FOR I=1 TO 3 STEP 2:PRINT D$(0,I):INPUT D$(A,
I):NEXT:D$(A,2)="99":D$(A,6)=D$(A-1,6)
  390 S#=0:W#=0:FOR I=1 TO A-1:IF D$(I,2)<>"99" THE
N S#=S#+VAL(D$(I,4)):W#=W#+VAL(D$(I,5))
  400 NEXT:D$(A,4)=STR$(S#):D$(A,5)=STR$(W#):GOTO 1
20
  410 CLS:PRINT "DATA KILL"
  420 INPUT "NO.";B:FOR I=0 TO 4:PRINT D$(0,I):PRIN
T D$(B,I):NEXT:INPUT "OK (Y)";Y$
  430 IF Y$<>"Y" AND Y$<>"y" THEN 120
```

```
  440 FOR I=B TO A-1
  450 FOR J=0 TO 6:D$(I,J)=D$(I+1,J):NEXT:NEXT:A=A-
1:GOTO 120
  460 CLS:PRINT "SAVE ":INPUT "DISK OK(Y)";Y$:IF Y$
<>"y" AND Y$<>"Y" THEN 120
  470 FILES:INPUT "FILE NAME ";F$
  480 OPEN F$ FOR OUTPUT AS #1
  490 FOR I=1 TO A:FOR J=0 TO 6:PRINT #1,D$(I,J):NE
XT:NEXT:FOR I=0 TO 9:PRINT #1,"*":NEXT:CLOSE:GOTO 1
20
```

EXAMPLE

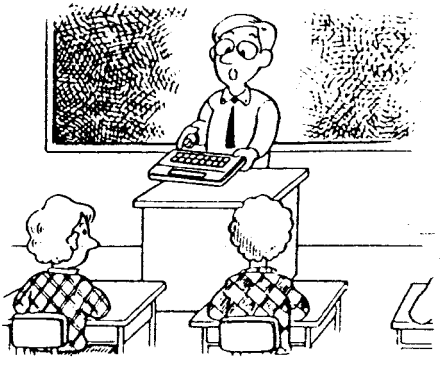
```
Example
                    EDIT                Total
MENU                Year,Month,Day      Year,Month,Day
1>>>Write           830804              830804
2>>>EDIT                                Memo
3>>>Sub-total                           DEF
4>>>Total           No.1
5>>>Kill                                DATA KILL
6>>>Save            DATA MAKE           NO.
Which ?             NO.2
                    Year,Month,Day
Write               830804              SAVE
NEW DATA            Memo                DISK OK(Y)
NO.1                ABC                 FILE NAME
END(Y)              From,Thru (No.)
```

DEBUG

1. If you wish to use a file name other than "A" as the FILE NAME, replace the "A" in the 4-line program described in the beginning of the text with the desired file name.
2. The code is always '99.'
3. The result of a sub-total or total operation is output to the position of the last data entered.

GRADING EXAMINATION PAPERS

This program permits calculating total grades, sorting records of performance by subject or some other key, and correcting input data.



rogram Outline

is extremely time-consuming work for teachers grade examination papers and calculate total d average grades. If the teacher makes a stake while tabulating examination grades, he she) will have to repeat the work from the y beginning.

re is a powerful aid for all teachers who must de examination papers from time to time. ter the input of individual grades, this pro- m immediately outputs the total grades, rage grades, ranking, etc., for each student.

ata Input

en the program is started, you are requested to input the number of students, the mber of subjects, and the student names. After this data has been input, the program uests you to input the subject names. When the subject names are input, the prompt NFORMATION?" is displayed. If you wish to provide additional information (e.g., ysical data), input the data. Otherwise, press the ↵ key.

xt, input the individual student grades by subject. The type of processing to be formed must now be specified. The program can perform the following types of cessing:

1. Edit
   Used to correct input data (marks or information).
   To correct input data, input the ID number of the relevant student.
2. Sort I
   Used to arrange the student names in total grade order (from highest to lowest).
3. Print
   Outputs total and average grades by subject for each student, as well as the average of total scores of all students.
4. Sort II
   Used to sort data according to a given key (information). For example, if roll numbers have been input, the student names are arranged in ascending order. If the student names have been input in upper-case letters, they are arranged in alphabetical order. The results of sorting may be printed.
5. Sort III
   Outputs the rankings of students by subject, together with their grades.

e that an edit operation after execution of a sort becomes complex since the ID nbers have been ignored. This sort of problem can be alleviated by inputting dent names in advance in the order of their roll numbers.

OUTPUT (Output data using LPRINT command)

```
Jhoson
Mathmatics      48
History         55
Physics         63
Smith
Mathmatics      88
History         33
Physics         56
Tomson
Mathmatics      37
History         78
Physics         95
Rogers
Mathmatics      98
History         14
Physics         35
Raike
Mathmatics      26
History         84
Physics         84
Anderson
Mathmatics      33
History         25
Physics         24
Total           Average
Jhoson
 166            55.3333
Smith
 177            59
Tomson
 210            70
Rogers
 147            49
Raike
 194            64.6667
Anderson
 82             27.3333
SD= 167.806
```

```
Tomson
Mathmatics      37
History         78
Physics         95
Raike
Mathmatics      26
History         84
Physics         84
Smith
Mathmatics      88
History         33
Physics         56
Jhoson
Mathmatics      48
History         55
Physics         63
Rogers
Mathmatics      98
History         14
Physics         35
Anderson
Mathmatics      33
History         25
Physics         24
Total           Average
Tomson
 210            70
Raike
 194            64.6667
Smith
 177            59
Jhoson
 166            55.3333
Rogers
 147            49
Anderson
 82             27.3333
SD= 167.806
```

LIST

1778 Bytes

```
 10 '
 20 '
 30 CLEAR:PRINT "GRADING EXAMINATION PAPER"
 40 INPUT "No. of students";N:INPUT "No. of subje
    cts";K
 50 DIM N$(N),K$(K),J$(N),T(N,K+2),S(N),B(N),H(N)
 60 FOR I=1 TO N
 70 PRINT I;:INPUT "th name";A$:N$(I)=LEFT$(A$,
   10):B(I)=I
 80 NEXT
 90 FOR I=1 TO K
100 PRINT I;:INPUT "Name of subject";K$(I)
110 NEXT
120 FOR I=1 TO N
130 INPUT "INFORMATION";J$(I):NEXT
140 FOR J=1 TO K:PRINT K$(J)
150 FOR I=1 TO N:PRINT "Scores of";N$(I)
160 INPUT T(I,J):NEXT:NEXT
180 FOR I=1 TO N:S=0:FOR J=1 TO K:S=S+T(I,J):NEXT
   :T(I,K+1)=S:T(I,K+2)=S/K:NEXT
190 INPUT "Task 1-5";T:ON T GOTO 210,260,350,600,
   800
200 GOTO 190
210 CLS:PRINT "EDIT ":INPUT "Number";B
220 PRINT B(B)
230 FOR I=1 TO K:PRINT K$(I);" ";T(B,I):INPUT "Sc
   ores";T(B,I)
240 NEXT:PRINT J$(B):INPUT "INFORMATION";J$(B)
250 GOTO 180
260 FOR I=1 TO N-1
270 FOR J=I+1 TO N
280 IF T(I,K+1)<T(J,K+1) THEN GOSUB 2000
290 NEXT:NEXT
340 GOTO 190
350 CLS:PRINT "PRINT "
360 S=0:FOR I=1 TO N
370 S=S+T(I,K+1)
380 NEXT
390 X=X/N '
400 S=0:FOR I=1 TO N:S=((T(I,K+1)-X)^2)+S
```

```
 410 NEXT
 420 SD=SQR(S/N)
 430 FOR I=1 TO N
 440 PRINT N$(I)
 450 FOR J=1 TO K:PRINT K$(J),
 460 PRINT T(I,J)
 470 IF INKEY$="" THEN 470
 480 NEXT:NEXT
 490 PRINT "Total","Average"
 500 FOR I=1 TO N:PRINT N$(I):PRINT T(I,K+1),T(I,K
     +2)
 505 IF INKEY$="" THEN 505
 510 NEXT:PRINT "SD=";SD:GOTO 190
 600 FOR I=1 TO N-1
 610 FOR J=I+1 TO N
 620 IF J$(I)<J$(J) THEN GOSUB 2000
 630 NEXT:NEXT:GOTO 190
 800 FOR O=1 TO K:PRINT K$(O)
 810 FOR I=1 TO N-1
 820 FOR J=I+1 TO N
 830 IF T(I,O)<T(J,O) THEN GOSUB 2000
 840 NEXT:NEXT:FOR I=1 TO N:PRINT N$(I);T(I,O)
 850 IF INKEY$="" THEN 850
 860 NEXT:NEXT:GOTO 190
2000 FOR L=1 TO K+2
2010 A=T(I,L):T(I,L)=T(J,L):T(J,L)=A:NEXT
2020 A=B(I):B(I)=B(J):B(J)=A:A$=J$(I):J$(I)=J$(J):
     J$(J)=A$:A$=N$(I):N$(I)=N$(J):N$(J)=A$
2030 RETURN
```

DEBUG

If an error occurs in any of the routines Edit, Sort I, Sort II, Sort III, or Print, check the appropriate routine. The Edit routine is in lines 210 to 250; the Sort I routine is in lines 260 to 340; the Print routine is in lines 350 to 510; the Sort II routine is in lines 600 to 630; and the Sort III routine is in lines 800 to 860.

SORTING/SUMMING

Program Outline

his program permits a retailer, etc., to check a ay's sales by sorting and totaling the sales by em. The results can be output on a printer. It is so possible to save the output data on a tape r printout at a later time.

Data Input

hen the program is started, it asks whether the data is to be input from the keyboard from a tape. If you select input from tape, data previously stored on tape is input. If ou select input from the keyboard, it is necessary to input the following data from e keyboard:

Code No.	Any number from 1 to 50.
Item name	Item name associated with Code No.
Quantity	Quantity sold.
Unit price	Unit price of the item.

ter inputting all the data, input Code No. 99. The program then asks whether the ta is to be saved on a tape. If you wish to save the data on tape, press the Y and ↵ ys in that order.

Output of Result

ter the data is input from the keyboard or tape, the program asks you to specify the vice on which the results are to be output.

1 DISPLAY:	The results are output on the display.
2 PRINTER:	The results are output on the printer.
3 DIS & PRI:	The results are output on both the display and the printer.

hen 1 (DISPLAY) is selected, the program displays the code, amount, and per- ntage of the total amount of the first item on the display as shown in the example of play ❶. In this case, the item name is not displayed due to the limitation on the mber of characters that can be displayed on a single line on the display screen. Press- any key on the keyboard causes the code, amount and percentage of the total ount of the next item to be displayed. The total amount is displayed at the end of s routine.

en 2 (PRINTER) is selected, the results are printed as shown in the example of play ❷.

en 3 (DIS & PRI) is selected, the results are displayed on the display screen and at same time are output on the printer.

enever possible, use the REMOTE terminal when writing data to or reading data m tape, in order to ensure that the write or read operation is performed properly.

The item names may not exceed eight characters in length. If an item name consisting of more than eight characters is input, the first character or characters are lost. Note also that when extremely large numbers are input, they are converted into exponential notation (e.g., 1250000 = 1.25 x 10^6 = 1.25E + 06).

❶ DISPLAY

```
    Code    Item price    Percentage
      1          8280          0.66
      2        702000         55.81
```

❷ OUTPUT

```
Code   Item name   Item price   Percentage
  1          A        56088         5.63
  2          B        80730         8.11
  3          C       157320        15.80
  4          D       245565        24.66
  5          E       105652        10.61
  6          F       302220        30.35
  7          G        48063         4.83
-------------------------------------------
Total              $ 995638       100.00%
```

LIST

2096 Bytes

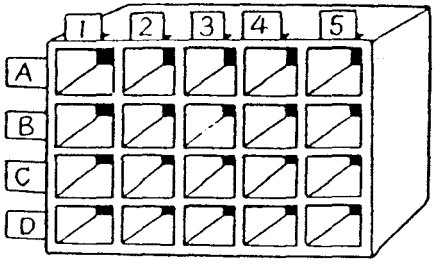
```
    10 '
    20 ' SORTING AND TOTALING
    30 '
    40 CLS:PRINT "SORTING AND TOTALING":CLEAR
    50 DIM DA$(50,1)
    60 PRINT "Method of data input"
    70 PRINT"< K >-- Keyboard":PRINT"< T >-- Tape":P
RINT"select K or T  "
    80 '
    90 Z$=INKEY$:IFZ$="K" OR Z$="k"THEN 100 ELSE IFZ
$="T" OR Z$="t" THEN 560 ELSE 90
   100 CLS:INPUT "Code No.";CD
   110 IF CD=99 THEN 200
   120 IF CD<=0 OR CD>50 THEN GOTO 100
   130 INPUT "Item name";S$:INPUT "Quantity";SU
   140 IF SU<1 THEN 130
   150 INPUT "Unite price";KI
```

```
 160 IF KI<=0 THEN 150
 170 H=VAL(DA$(CD,0)):DA$(CD,0)=STR$(SU*KI):DA$(CD
,1)=S$
 180 TL=TL+SU*KI-H
 190 GOTO 100
 200 INPUT"Do you wish to save data to tape (Y/N)"
Z$
 210 IF Z$="Y" OR Z$="y" THEN GOTO 680
 220 CLS:PRINT"Which device"
 230 PRINT
 240 PRINT "1>>DISPLAY"
 250 PRINT "2>>PRINTER"
 260 PRINT "3>>DISPLAY & PRINTER"
 270 INPUT KB
 280 IF KB<1 OR KB>3 THEN 270
 290 ON KB GOSUB 310,430,530
 300 END
 310 CLS
 320 PRINT "Code  Price  Percent"
 330 FOR I=1 TO 50
 340 IF DA$(I,0)="" THEN 390
 350 PS=VAL(DA$(I,0))/TL*100
 360 NO$=" "+STR$(I)
 370 PRINT RIGHT$(NO$,2);:PRINT USING"##########";
AL(DA$(I,0));:PRINT USING" ###.##";PS
 380 A$=INKEY$:IF A$="" THEN 380
 390 NEXT I
 400 T=TL/TL*100
 410 PRINT USING"Total$####### ###.##%";TL;T
 420 RETURN
 430 LPRINT "Code  Item name  Item price  Percenta
e"
 440 FOR I=1 TO 50
 450 IF DA$(I,0)="" THEN 490
 460 PS=VAL(DA$(I,0))/TL*100
 470 NO$="  "+STR$(I):DA$(I,1)="        "+DA$(I,1):
A$(I,0)="        "+DA$(I,0):W$=RIGHT$(NO$,3)+"     "
RIGHT$(DA$(I,1),8)+"    "+RIGHT$(DA$(I,0),8)
 480 LPRINT W$;"     ";:LPRINT USING"###.##";PS
 490 NEXT I
 500 LPRINT"----------------------------------------
":LPRINT"Total            ";RIGHT$("                $
+STR$(TL),11);
 510 T=TL/TL*100:LPRINT USING"     ###.##%";T :FOR
=0 TO 5:LPRINT:NEXT
```

```
  520 RETURN
  530 GOSUB 310
  540 GOSUB 430
  550 RETURN
  560 'INPUT FROM TAPE
  570 CLS:PRINT"Input from tape. Press RETURN key w
hen tape is ready."
  580 IF INKEY$<>CHR$(13) THEN 580
  590 OPEN"CAS0:DATA"FOR INPUT AS #1
  600 FOR I=1 TO 50
  610 INPUT#1,I
  620 INPUT#1,DA$(I,1)
  630 INPUT#1,DA$(I,0)
  640 NEXT I
  650 INPUT#1,TL
  660 CLOSE #1
  670 GOTO 790
  680 'RECORDING TO TAPE
  690 CLS:PRINT"Save data to tape. Press RETURN key
 when tape is ready."
  700 IF INKEY$<>CHR$(13) THEN 700
  710 OPEN"CAS0:DATA"FOR OUTPUT AS #1
  720 FOR I=1 TO 50
  730 PRINT#1,I
  740 PRINT#1,DA$(I,1)
  750 PRINT#1,DA$(I,0)
  760 NEXT I
  770 PRINT#1,TL
  780 CLOSE #1
  790 CLS:LOCATE 3,3:PRINT"Stop tape.":FOR KK=1 TO
500:NEXT KK:GOTO 220
```

CROSS-TOTALING

Program Outline



There are various ways of cross totaling. This program is extremely useful in calculating subtotals of items in horizontal rows and in vertical columns. The data table prepared by this program can be immediately stored in a CETL file.

Data Input and Subtotal Output

When the program is started, it asks you to input the number of items in a horizontal row and the number of records in a vertical column. Input numbers from 2 to 24 for the number of items and the number of records, respectively. When all the numeric data in the first row has been entered, the program calculates the subtotal of that row. This processing is continued until the subtotal for the last row is calculated. The program then displays the prompt "DO YOU WANT TO CHANGE ANY ITEM?" If you do not wish to change any numeric data, input 0 for either X or Y. If you change numeric data in a particular row, the program re-calculates the subtotal of that row and outputs the result.

(Output of percentage data, etc.)

After all the data has been input, the program asks whether or not the results are to be output to the printer. If you wish to print out the results, press the Y key. If not, press any other key on the keyboard. When you press the Y key, the results are printed as shown in the example. Note that the maximum number of items (X) that can be printed in a row is 9 for the FP-1017PR printer and 3 for the FP-1011PL printer-plotter.

If you press anything other than the Y key, the program outputs the subtotals and percentages of the individual columns as you press the ↵ key repeatedly. Thus, percentage data is output only in vertical columns. This should be kept in mind when designing the input format.

Example of output

CROSS TOTAL

	X 1	X 2	X 3	X 4	X 5	Total
Y 1	37	29	46	89	29	230
Y 2	35	99	23	34	58	249
Y 3	73	72	34	14	94	287
Y 4	36	24	45	28	59	192
Y 5	83	14	95	36	74	302
Total	264	238	243	201	314	1260
Ratio%	21.0	18.9	19.3	16.0	24.9	——

This program uses the CETL file F0. Therefore, if this file is already occupied by some other data, it should be cleared by the K command before the program can be stored. Note also that inputting extremely large numbers (i.e., so large that a calculation result exceeds 8 digits) produces incorrect output.

LIST *2032 Bytes*

```
   10 'CROSS TOTALING
   20 'CROSS TOTAL
   30 CLS:CLEAR:PRINT"CROSS TOTALING":PRINT"Vertica
l columns 24,horizontal rows 24":PRINT" OK":PRINT
   40 INPUT"Number of Columns.X";X
   50 IF X>25 OR X<2 THEN 40
   60 INPUT"Number of rows.   Y";Y
   70 IF Y>25 OR Y<2 THEN 60
   80 PRINT:PRINT" X - Y"
   90 FOR J=1 TO Y+1:FL(0,X+1,J)=0:NEXT
  100 FOR I=1 TO X+1:FL(0,I,Y+1)=0:NEXT
  110 FOR J=1 TO Y
  120 FOR I=1 TO X
  130 PRINTI;"-";J;" ";:INPUT A
  140 FL(0,I,J)=A:FL(0,X+1,J)=FL(0,X+1,J)+A:FL(0,I,
Y+1)=FL(0,I,Y+1)+A:NEXT I
  150 PRINT"Sub-total of(Y=";J;")";FL(0,X+1,J):PRIN
T
  160 FL(0,X+1,Y+1)=FL(0,X+1,Y+1)+FL(0,X+1,J):NEXT
J
  170 PRINT"Input completed"
  180 PRINT:INPUT"Do you want to change any item ?
X=";Q:IF Q<0 OR Q>X THEN 180
  190 INPUT"Y=";W:IF W<0 OR W>Y THEN 190
  200 IF Q=0 OR W=0 THEN 280
  210 PRINTQ;"-";W;"=";FL(0,Q,W)
  220 PRINT"Do you wish to change numeric data?(Y/N
)"
  230 Z$=INKEY$:IF Z$="" THEN 230 ELSE IF Z$="Y" OR
 Z$="y" THEN 240 ELSE 180
  240 PRINTQ;"-";W;" ";:INPUT A
  250 FL(0,Q,Y+1)=FL(0,Q,Y+1)-FL(0,Q,W)+A:FL(0,X+1,
W)=FL(0,X+1,W)-FL(0,Q,W)+A
  260 FL(0,X+1,Y+1)=FL(X0,+1,Y+1)-FL(0,Q,W)+A:FL(0,
Q,W)=A
  270 PRINT"Sub-total of (Y=";W;")";FL(0,X+1,W):GOT
O 180
```

```
  280 PRINT:PRINT"Are the results to be output to p
rinter?      (Y OR N)"
  290 Z$=INKEY$:IF Z$="" THEN 290 ELSE IF Z$="Y" OR
 Z$="y" THEN 350
  300 FOR I=1 TO X:PRINT:PRINT"Sub-total of (X=";I;
")";FL(0,I,Y+1)
  310 PRINTUSING"Configuraton is     ##.#%";FL(0,I,Y
+1)/FL(0,X+1,Y+1)*100
  320 Z$=INKEY$:IF Z$="" THEN 320 ELSE NEXT
  330 PRINT:PRINT"Total of Y";FL(0,X+1,Y+1)
  340 END
  350 'PRINTER
  360 IF X>9 THEN PRINT"Data can't be output to pri
nter":GOTO 300
  370 CLS:LPRINT" <CROSS TOTAL>":LPRINT
  380 LPRINT"┌──────┬";:FOR I=1 TO X:LPRINT"──────┬
";:NEXT:LPRINT"──────────┐"
  390 LPRINT"│      ";:FOR I=1 TO X:LPRINT"│  X";I;
:NEXT:LPRINT"│  Total  │"
  400 FOR J=1 TO Y:GOSUB 460
  410 LPRINT"│  Y";J;:FOR I=1 TO X:LPRINT USING"│##
####";FL(0,I,J);:NEXT I:LPRINT USING"│#########│";FL
(0,X+1,J):NEXT J
  420 GOSUB 460:LPRINT"│ Total";:FOR I=1 TO X:LPRIN
T USING"│#######";FL(0,I,Y+1);:NEXT:LPRINT USING"│##
######│";FL(0,X+1,Y+1)
  430 GOSUB 460:LPRINT"│Ratio%";:FOR I=1 TO X:LPRIN
T USING"│  ##.#";FL(0,I,J)/FL(0,X+1,Y+1)*100;:NEXT:
LPRINT"│  ----  │"
  440 LPRINT"└──────┴";:FOR I=1 TO X:LPRINT"──────┴
";:NEXT:LPRINT"──────────┘"
  450 END
  460 LPRINT"├──────┼";:FOR II=1 TO X:LPRINT"┼──────
";:NEXT II:LPRINT"┼──────────┤":RETURN
```

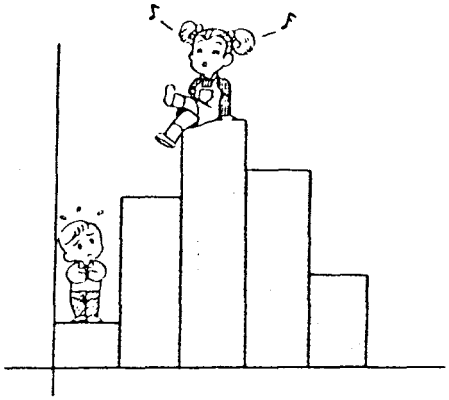
DEBUG

f no printer is available, lines 280, 290, and 350 to 460 may be omitted. In this ase, change THEN 280 in line 200 to THEN 300. The number of items in a hori- ontal row and/or the number of records in a vertical column can be increased by hanging the number 24 in lines 30, 50, and 70 to the desired number. In this ase, do not forget to change the number of items and/or the number of records CETL accordingly.

FREQUENCY DISTRIBUTION TABLE

Program Outline

The frequency distribution table is a table showing the number of occurrences of each of several classes which represent the range of a given variable. A representation of a frequency distribution by means of rectangles whose widths represent classes and whose heights represent the corresponding frequencies is called a histogram. This program outputs a frequency distribution table and the associated histogram on a plotter-printer (FP-1011PL).



Data Input

1. Starting the program

When the program is started, it displays some simple explanatory messages. Press any key on the keyboard; the program asks for the method of data input. If you wish to input data from the keyboard, press the K key. If you wish to input data from tape, press the T key.

2. Inputting data from the keyboard

If you press the K key, input the title first. Because of space limitations, the title must not exceed 15 characters in length. The program then displays the prompt "NO.1 ——>." Input data in response to the prompt.

If incorrect data is input, press the "—" key and the program returns to the previous input. You can then input the correct data. Pressing the E key terminates data input.

During data input from the keyboard, press the CTRL and N keys simultaneously after the title is input. This makes it possible to use the central section of the keyboard as a ten-key keyboard, facilitating the input of numeric data. Pressing the CTRL and N keys again resets the keyboard to its normal function.

3. Inputting data from tape

Set the REMOTE switch on the FP-200 to ON.

After pressing the T key, connect the tape recorder, set it for playback and press the ↵ key. The input of data from the tape will start.

4. Correcting the data

At the end of data input from the keyboard or tape, the program displays the title and the number of data items. It then asks whether or not there is data to be corrected. If there is data to be corrected, press the Y key. If not, press the N key. To correct data, first input the data number to be corrected, then input the correct data. Repeat this operation until all the data is correct. Press the N key when there is no more data to be corrected.

Outputting data

ter all the corrections are made, the program asks for the number of classes. Input a mber from 2 to 10; the program starts processing the input data. Eventually, the ogram displays the maximum value, minimum value, class widths, and highest equency on the display screen, and outputs the frequency distribution table and stogram on the plotter-printer.

Recording data

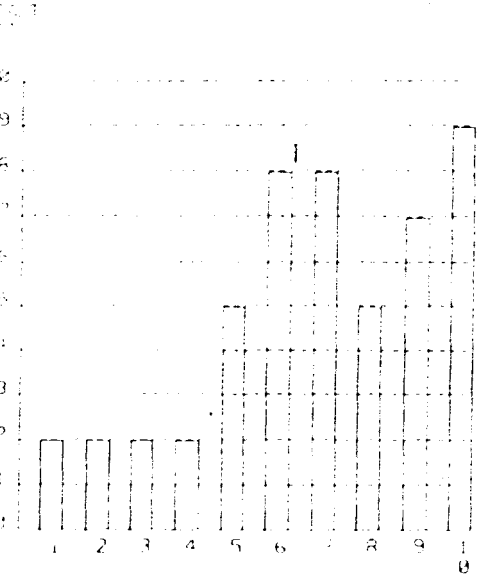
ter the printout, the program asks whether the output data is to be saved on tape. you wish to save the data on tape, press the Y key. If not, press the N key.

ter pressing the Y key, set the tape recorder for recording and press the ↵ key. ecorded on the data tape are the title name, number of data items, and values of data ms, under the file name "DOSDATA."

ample of data input

No.=	1	78	No.= 11	66	No.= 21	90	No.= 31	71	No.= 41	64
No.=	2	7	No.= 12	98	No.= 22	88	No.= 32	60	No.= 42	56
No.=	3	67	No.= 13	34	No.= 23	23	No.= 33	52	No.= 43	98
No.=	4	12	No.= 14	55	No.= 24	45	No.= 34	80	No.= 44	23
No.=	5	45	No.= 15	60	No.= 25	56	No.= 35	90	No.= 45	39
No.=	6	34	No.= 16	76	No.= 26	78	No.= 36	96	No.= 46	49
No.=	7	38	No.= 17	87	No.= 27	89	No.= 37	55	No.= 47	65
No.=	8	87	No.= 18	67	No.= 28	89	No.= 38	45	No.= 48	67
No.=	9	67	No.= 19	87	No.= 29	85	No.= 39	91	No.= 49	87
No.=	10	56	No.= 20	54	No.= 30	67	No.= 40	74	No.= 50	92

mple of printout



* FREQUENCY DISTRIBUTION *

Class	Range		Frequency
1	7.000 ~	16.100	7
2	16.100 ~	25.200	7
3	25.200 ~	34.300	7
4	34.300 ~	43.400	7
5	43.400 ~	52.500	5
6	52.500 ~	61.600	6
7	61.600 ~	70.700	8
8	70.700 ~	79.800	[illegible]
9	79.800 ~	88.900	[illegible]
10	88.900 ~	98.000	[illegible]
		Total	50

Number of data [illegible] 50
Maximum value [illegible] 98.000
Minimum value [illegible] 7.000

7. Terminating the program
The program now asks whether or not the same input data is to be processed using
different number of classes. If you wish to do so, press the Y key. If not, press th
N key.
After the Y key, has been pressed, the program returns to step 5 (outputting data).
When the N key is pressed, the program is terminated. To process new data, re-execut
the program.

This program does not contain a routine to display and print input data. Since inpu
data is stored in the array variable IN, the entire data can be output on a plotter b
executing the LIST 2 program in a separate program area (execute the PROG command).

LIST *3791 Bytes*

```
LIST 1
  100 CLEAR1024:OPTION BASE 1:DIM IN(100),DS(10)
  110 T$="** FREQUENCY                DISTRIBUTION **"
  120 ML$="":FOR I=1 TO 37:ML$=ML$+"-":NEXT I:E$=CH
R$(27)
  130 '
  140 CLS:LOCATE 0,2:PRINT T$:FOR I=0 TO 800:NEXT
  145 CLS:PRINT"Frequency distribut-ion table is cr
eated from input data."
  148 PRINT"Spacify number of   classes. Data can b
e saved to cassette  tape."
  149 PRINT" << Hit any key >>";
  150 IF INKEY$=""THEN 150
  200 '========== Input data
  210 CLS:PRINT T$:PRINT"Method of data input";
  220 PRINT"< K >-- keyboard":PRINT"< T >-- tape":P
RINT"select K or T ";
  230 Z$=INKEY$:IFZ$="K" OR Z$="k" THEN 300 ELSE IF
 Z$="T" OR Z$="t" THEN 400 ELSE 230
  300 '========== Input from keyboard
  310 CLS:PRINT T$
  320 INPUT"Title :",A$:PRINT:C=0
  330 Z$=STR$(IN(C+1)):INPUT "No."+STR$(C+1)+"-->",
Z$:IFZ$="E"THEN 500
  340 IF Z$="-" THEN IF C<1 THEN 330 ELSE C=C-1:GOT
O330
  350 C=C+1:IN(C)=VAL(Z$):IF C<100 THEN 330 ELSE 50
0
  400 '========== Input from tape
```

```
  410 CLS:PRINT T$:PRINT"Input data from tape. Pres
s RETURN key   when tape is ready."
  420 IF INKEY$<>CHR$(13) THEN 420
  430 OPEN "CAS0:DOSDATA" FOR INPUT AS #1
  440 INPUT #1,A$,C
  450 FOR I=1 TO C:INPUT #1,IN(I):NEXT I:CLOSE #1
  500 '========== Correction
  510 CLS:PRINT T$
  520 PRINT"Title :";A$:PRINT"Number of data:";C
  530 PRINT"Do you wish to corr-ect data.";
  540 Z$=INKEY$:IFZ$="N" OR Z$="n" THEN 600 ELSE IF
 Z$="Y" OR Z$="y" THEN PRINTZ$:GOTO 550ELSE GOTO 54
0
  550 INPUT"Data No. :",Z:IF Z<1 OR Z>C THEN 550
  560 PRINT"   ";IN(Z);:INPUT"-->",IN(Z):PRINT:GOTO
 530
  600 '========== Caliculation 1
  610 PRINTZ$:SUM=0:MAX=-1E99:MIN=1E99
  620 FOR I=1 TO C:PRINT"*";:SUM=SUM+IN(I):IF MAX<I
N(I) THENMAX=IN(I)
  630 IF MIN>IN(I) THEN MIN=IN(I)
  640 NEXT I:PRINT
  700 '========== Input number of classes
  710 CLS:PRINTT$;:PRINT"< ";A$:PRINT"Number of dat
a ":PRINTC
  720 PRINT"Number of class  ":INPUT"";Z$:IF Z$="-"
THEN 500
  730 CL=VAL(Z$):IF CL<2 OR CL>10 THEN710
  800 '========== Frequency count
  810 LE=(MAX-MIN)/CL
  820 CLS:PRINT"Maximum value ":PRINTMAX:PRINT"Mini
mum value ":PRINTMIN
  823 PRINT:PRINT "<< Hit any key >>"
  824 IF INKEY$="" THEN 824
  825 CLS:PRINT"Class width ":PRINTLE
  830 FOR I=1 TO 10:DS(I)=0:NEXT I
  840 FOR I=1 TO C:Z=INT((IN(I)-MIN)/(LE-1E-6)):IF
Z<CL THEN Z=Z+1
  850 DS(Z)=DS(Z)+1:NEXT I
  860 M=DS(1)
  870 FOR I=2 TO CL:IF M<DS(I) THEN M=DS(I)
  880 NEXT I
  890 PRINT"highest frequency ":PRINTM
  900 '========== Caliculation 2
  910 N=M:X=0:I=1
```

```
  920 IF N>100 THEN N=INT(N/10):X=X+1:GOTO 920
  930 IF N MOD 2>0 THEN N=N+1
  940 IF N MOD I>0 OR N/I>10 THEN I=I+1:GOTO 940
  950 N=N*10^X:IF ABS(10-I)>ABS(10-N/I) THEN E=N/I
ELSE E=I
 1000 '========== Frequency distribution table
 1010 LPRINTCHR$(28);CHR$(37):LPRINT"Y0":LPRINT"Z2,
6":LPRINT"Q0":LPRINT"B2":LPRINT"L0":LPRINT"J0"
 1020 LPRINT"S2":LPRINT"P";A$:LPRINT"S1":LPRINT"O10
,-90"
 1030 LPRINT"D0,80,0,0,80,0"
 1040 SC=80/E:T=N/80:IF LEN(STR$(M))>4 THEN S=0:B=7
 ELSE S=1:B=3
 1050 LPRINT"S";S:LPRINT"L1"
 1060 FOR I=E TO 0 STEP -1:Y=I*SC:N$=RIGHT$("     "+
STR$(FIX(Y*T)),B):LPRINT"M-9,";Y:LPRINT"P";N$:IF I<
>0 THEN LPRINT"D0,";Y;",80,";Y
 1070 NEXT I:LPRINT"S1":LPRINT"J1":LPRINT"L0":SC=80
/CL
 1080 FOR I=1 TO CL:X2=I*SC:X1=X2-SC/2:Y=80/N*DS(I)
:IF Y<0.1 THEN Y=0
 1090 Z$=MID$(STR$(I),2):YM=-4:FOR J=1 TO LEN(Z$):L
PRINT"M";X2-SC/4;",";YM:LPRINT"P";MID$(Z$,J,1):YM=Y
M-3.6:NEXT J
 1100 LPRINT"D";X1;",0,";X1;",";Y;",";X2;",";Y;",";
X2;",0"
 1110 NEXT I:LPRINT"H10":LPRINT"J0"
 1200 '========== Table
 1210 LPRINTCHR$(28);CHR$(46)
 1220 LPRINTE$;"S2":LPRINT MID$(T$,2,11);TAB(12);MI
D$(T$,26,14):LPRINTE$;"S1"
 1230 LPRINT"Class";TAB(13);"Range ";TAB(28);"Frequ
ency":LPRINTML$
 1240 FOR I=1 TO CL:LS=MIN+LE*(I-1):MS=LS+LE:LPRINT
USING"###  ######.### ~ ######.###  #####";I;LS;MS;
DS(I):NEXT I
 1250 LPRINT ML$:LPRINT TAB(25);"Total";:LPRINT USI
NG "#####";C
 1260 LPRINT E$;"F3"
 1270 LPRINT USING"Number of data: ######";C
 1280 LPRINT USING"Maximum value : ######.###";MAX
 1290 LPRINT USING"Minimum value : ######.###";MIN:
LPRINT E$;"F5"
 1300 '========== Save data on tape
 1310 CLS:INPUT"Do you save data on tape?";Z$
```

```
 1315 IFZ$="Y"ORZ$="y"THEN1320ELSEIFZ$="N"ORZ$="n"T
HEN1400ELSE1310
 1320 CLS:PRINT T$:PRINT"Data is saved to tape. Pre
ss RETURN key when tape is ready"
 1330 Z$=INKEY$
 1340 IF Z$<>CHR$(13) THEN 1330
 1350 OPEN "CAS0:DOSDATA" FOR OUTPUT AS #1
 1360 PRINT#1,A$;",";C
 1370 FOR I=1 TO C:PRINT #1,IN(I):NEXT I
 1380 CLOSE #1
 1400 CLS:INPUT"Do you change the n-umber of class"
;Z$
 1410 IF Z$="Y" OR Z$="y"THEN 700 ELSEIF Z$="N" OR
Z$="n"THEN END ELSE1400
```

LIST 2

```
   10 PRINT"Printr Ready <Push any key>"
   20 IF INKEY$="" THEN 20
   30 FOR I=1 TO C
   40 LPRINT USING"No.=###   ######";I;I(I)
   50 NEXT I
   60 END
```

DEBUG

When the number of data items exceeds 100, data input is automatically terminated. To prevent this, change the subscript for the array variable IN in the DIM statement in line 100 (the number 100 in parentheses) and the value of the right side of the IF conditional statement in line 350 to the appropriate number of data items.

A 'MOUNT error' may occur when reading data from, or writing data to, a cassette tape. This occurs if a greater number of external files than the preset number of buffers was opened.

Since the number of buffers is set to 0 after a RESET, the processing of a file must be performed after reserving a file buffer by executing a MOUNT statement in direct mode. This program has only one file to be opened. Hence, the 'MOUNT error' problem can be prevented by executing MOUNT 1.

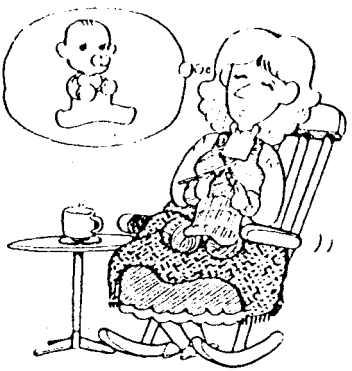
The number of buffers set by the MOUNT statement is retained even after the power is turned off.

Note, however, that executing a MOUNT statement erases all current variables, requiring re-input of the appropriate data.

NUMBER OF DAYS CALCULATION

Program Outline

This program calculates the number of days from a certain date (year, month, day) until another specified date (year, month, day), or conversely, determines the date (year, month, day) a specified number of days after a certain date (year, month, day). The former operation is called the 'number of days calculation,' and the latter is called the 'inverse number of days calculation.'



Directions

When the program is started, it asks whether or not the results are to be output on a printer. If you wish to print out the result, press the Y key. If not, press the N key.
Next, specify either the 'number of days calculation' or the 'inverse number of days calculation'.

* Number of days calculation (select '1')

First input the starting year, month and day. Next, input the ending year, month and day in response to the prompt from the program. The program then displays the number of days, together with the starting and ending dates and their days of the week. It is also possible to calculate the number of days back to a past date. In this case, the result is output prefixed with a – (minus) sign.

* Inverse number of days calculation (select '2')

First, input the starting date (year, month, day). Next, input the number of days. (If the number of days is to be reckoned back to a past date, put a – (minus) at the left of the number. In this case, the result appears as 'the date the specified number of days before the starting date is xxxx.')
Pressing any key on the keyboard returns to the selection of the type of calculation. Pressing the BREAK key terminates program execution.

EXAMPLE

```
Select '1' NUMBER OF DAYS CALCULATION
From 1965 6th-month 21th-day Monday
  to 1983 11th-month 12th-day Saturday
-----------------------------------
 6718days

select '2' INVERSE NUMBER OF DAYS CALCULATION
From 1962 6th-month 7th-day Thursday
 1110days after
-----------------------------------
 on  1965 6th-month 21th-day Monday
```

L I S T *2701 Bytes*

```
 100 '
 110 CLEAR 1024:DIM B(3),G$(3),YO$(6)
 120 GOSUB 1000
 130 '
 140 CLS:PRINT T$
 150 PRINT "Printer (Y/N) :"
 160 Z$=INKEY$:IF Z$="Y" OR Z$="y" THEN PSW=1:GOTO
180
 170 IF Z$="N" OR Z$="n" THEN PSW=0 ELSE GOTO 160
 180 '========== Menu
 190 CLS:PRINT T$:PRINT
 200 PRINT " 1 )Number of days          calculatio
":PRINT
 210 PRINT " 2 )Inverse                  calculato
"
 220 Z$=INKEY$:IF Z$="1" THEN 240
 230 IF Z$="2" THEN 390 ELSE GOTO 220
 240 '========== NUMBER OF DAYS CALCULATION
 250 CLS:PRINT "* Number of days          calcula
ion *"
 260 GOSUB 660
 270 GOSUB 710
 280 NS=EN-SN
 290 GOSUB 960
 300 GE$=EG$:Y=EY:Z$=STR$(Y)
 310 Z3$=" to "+Z$+STR$(EM)+"th-month":Z4$=STR$(ED
+"th-day "+YO$(EU)
 320 Z5$=STR$(NS)+"days"
 330 IF PSW=0 THEN 360
 340 LPRINT Z1$;Z2$:LPRINT Z3$;Z4$
 350 LPRINT T2$:LPRINT Z5$:GOTO 180
 360 CLS:PRINT Z1$:LOCATE 2,1:PRINT Z2$
 370 LOCATE 0,3:PRINT Z3$:LOCATE 2,4:PRINTZ4$:LOCA
E 5,6:PRINT Z5$
 380 IF INKEY$="" THEN 380 ELSE GOTO 180
 390 '========== INVERSE NUMBER OF DAYS CALCULATIO

 400 CLS:PRINT "*Inverse number of     days calcul
:ion*"
 410 GOSUB 660
```

```
  420 INPUT "Days    :",IN
  430 NS=SN+IN:Y=INT(NS/365)
  440 M=1:D=0:GOSUB 850:AN=NS-N
  450 IF AN>365 THEN SW=1:GOSUB 940
  460 IF AN<1 THEN SW=-1:GOSUB 940
  470 IF Y MOD 4>0 OR Y MOD 100=0 AND Y MOD 400>0 T
HEN SW=0 ELSE SW=1
  480 I=0:RESTORE 1080
  490 READ X:IF AN<=X THEN 530
  500 IF I=1 AND SW=1 THEN X=X+1
  510 AN=AN-X:I=I+1:GOTO 490
  520 '
  530 GY=Y:GM=I:GD=AN:GU=NS MOD 7
  540 GOSUB 960
  550 Z3$=STR$(ABS(IN))
  560 IF IN>0 THEN Z$="days after" ELSE Z$="days be
fore"
  570 Z3$=Z3$+Z$
  580 GE$=SG$:Y=GY:Z$=STR$(Y)
  590 Z4$=" on "+Z$+STR$(GM)+"th-month":Z5$=STR$(GD
)+"th-day "+YO$(GU)
  600 IF PSW=0 THEN 630
  610 LPRINT Z1$;Z2$:LPRINT Z3$
  620 LPRINT T2$:LPRINT Z4$;Z5$:GOTO 180
  630 CLS:PRINT Z1$:LOCATE 2,1:PRINT Z2$
  640 LOCATE 3,3:PRINT Z3$:LOCATE 0,5:PRINT Z4$:LOC
ATE 2,6:PRINT Z5$
  650 IF INKEY$="" THEN 650 ELSE GOTO 180
  660 '========== Start Year,Month,Day
  670 PRINT "Start Year,Month,Day"
  680 GOSUB 760:SY=Y:SM=M:SD=D
  690 GOSUB 850:SN=N:SU=U
  700 RETURN
  710 '========== End   Year/Month/Day
  720 PRINT :PRINT"End   Year,Month,Day"
  730 GOSUB 770:EY=Y:EM=M:ED=D
  740 GOSUB 850:EN=N:EU=U
  750 RETURN
  760 '========== Input Year,Month,Day
  770 Z=0
  780 INPUT "      :",Y,M,D
  790 Y=Y+B(Z)
  800 IF Y<1583 OR M<1 OR M>12 THEN 780
  810 IF M=12 THEN X=31:GOTO 830
  820 RESTORE 1080:FOR I=1 TO M+1:READ X:NEXT I
```

```
  830 IF D<1 OR D>X THEN 780
  840 RETURN
  850 '========== Nomber of days calculation
  860 U=Y-1
  870 N=U*365+INT(U/4)-INT(U/100)+INT(U/400)+D
  880 RESTORE 1080
  890 FOR I=1 TO M:READ X:N=N+X:NEXT I
  900 IF Y MOD 4>0 OR Y MOD 100=0 AND Y MOD 400>0 T
HEN 920
  910 IF M>2 THEN N=N+1
  920 U=N MOD 7
  930 RETURN
  940 '========== Year
  950 Y=Y+SW:AN=AN-365*SW:RETURN
  960 '==========
  970 GE$=SG$:Y=SY:Z$=STR$(Y)
  980 Z1$="From"+Z$+STR$(SM)+"th-month":Z2$=STR$(SD
)+"th-day "+YO$(SU)
  990 RETURN
 1000 '==========
 1010 T$="*Days calculation*"
 1020 T1$="":FOR I=1 TO 37:T2$=T2$+"-":NEXT I:
 1030 FOR I=0 TO 6:READ YO$(I):NEXT I
 1040 RETURN
 1050 '
 1060 DATA Sunday,Monday,Tuesday,Wednesday,Thursday
,Friday,Saturday
 1070 '
 1080 DATA 0,31,28,31,30,31,30,31,31,30,31,30
```

DEBUG

The data in line 1610 indicates the number of days in each of the twelve months. The adjustment for leap years is made by lines 580, 1120, 1150 to 1160.

PERPETUAL CALENDAR

Program Outline

This program provides a perpetual calendar, which is useful for trip or work planning. It is capable of displaying the day of the week of a particular day of a particular year, either in the past or future. It can, of course, produce a calendar for any past year.
By using a plotter, it is possible for the program to print a calendar in any of three different character sizes.

Data Input

When the program is started, it asks whether or not a plotter will be used. If you use a plotter, press the Y key. If not, press the N key.
When you press the N key, the program immediately outputs a calendar for the specified year/month on the display screen.
When you press the Y key, the program displays the prompt "CHAR. SIZE (0–2): –," asking the size of characters to be output on the plotter. The sizes of characters that can be specified are as follows:

0 25 mm x 20 mm (horizontal x vertical)
1 50 mm x 40 mm
2 75 mm x 60 mm

Input a number (0–2) and the year and month whose calendar is desired. The program then outputs a 4-color calendar on the plotter and a monochrome calendar on the display screen.

Example of character sizes and calendar output

Character size '0'

1983. 7

Su	Mo	Tu	We	Th	Fr	Sa
*	*	*	*	*	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	*	*	*	*	*	*

Character size '1'

1983. 7

Su	Mo	Tu	We	Th	Fr	Sa
*	*	*	*	*	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	*	*	*	*	*	*

Character size '2'

1983. 7

Su	Mo	Tu	We	Th	Fr	Sa
*	*	*	*	*	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	*	*	*	*	*	*

e calendar currently used throughout the world is called the Gregorian calendar, roduced by Pope Gregory XIII in October of 1582. Before that, the calendar had en subject to an error of one day every 128 years. For this reason, this program is igned such that it does not accept year data in or before 1582. The Gregorian endar is quite accurate since it eliminates the above-mentioned error by providing a leap year of 366 days every fourth year except for century years (which are leap rs only if they are divisible by 400).

LIST *1816 Bytes*

```
    1 '************************************
    2 '
    3 ' Calender (FP-200 & FP-1011PL)
    4 '
    5 '          copyright 1983 A.Kakizono
    6 '
    7 '          1983/ 6/11
    8 '
    9 '************************************
  100 '
  110 CLEAR 1023:DIM L$(6)
  120 L$(0)="Su Mo Tu We Th Fr Sa"
  130 '
  140 CLS:PRINT "* CALENDAR..FP-200 *"
  150 INPUT " Printer (Y/N) :",Z$:PSW=0:IF Z$="Y" O
R Z$="y" THEN PSW=1 ELSE 200
  160 INPUT "Char. size(0~2):",CS:IF CS<0 OR CS>2 T
HEN 160
  200 '
  210 PRINT:INPUT "Year  :",N:S=0
  220 IF N<1583 THEN 210
  230 INPUT "Month :",T:IF T<1 OR T>12 THEN 230
  240 '
  250 Y=N:Z=T:GOSUB 400:A1=X
  260 Y=N:Z=T+1:GOSUB 400:A2=X
  270 FOR J=1 TO 6:L$(J)="":NEXT J:GOSUB 500:GOSUB
600
  280 IF PSW=1 THEN GOSUB 700:PRINT:GOTO 200
  290 IF INKEY$="" THEN 290 ELSE PRINT:GOTO 200
  400 '
  410 IF Z<3 THEN Y=Y-1:Z=Z+12
```

```
  420 U=INT(Y/100):V=Y-U*100:X=INT(U/4)-U*2+V+INT(V
/4)+INT((Z+1)*26/10):X=X-INT(X/7)*7
  430 RETURN
  500 '
  510 A=A2-A1:IF A1<=A2 THEN A=A+28 ELSE A=A+35
  520 B=A1:C=0:IF S=1 THEN M$="S"+STR$(N-1925)+CHR$
(242)+STR$(T)+CHR$(243) ELSE M$=STR$(N)+"."+STR$(T)
  530 B1=A1:B2=A2:I=1:IF B1<>0 THEN FOR J=1 TO B1:L
$(I)=L$(I)+"  *":NEXT J
  540 FOR J=B TO 6:C=C+1:L$(I)=L$(I)+RIGHT$(" "+STR
$(C),3):IF A=C THEN 560
  550 NEXT J:L$(I)=MID$(L$(I),2):B=0:I=I+1:GOTO 540
  560 IF B2=0 THEN 580
  570 FOR J=1 TO 7-B2:L$(I)=L$(I)+"  *":NEXT J:L$(I
)=MID$(L$(I),2)
  580 RETURN
  600 '
  610 CLS:FOR J=0 TO I:PRINT L$(J);:NEXT J:PRINT "
   <";M$;" >";
  620 INIT(0,0),1,1:X=159:Y1=I*8+7
  630 DRAW (0,7)-(X,7)-(X,Y1)-(0,Y1)-(0,7)
  640 RETURN
  700 '
  710 X=(CS+1)*1.2:X1=X*9/2:X2=X*51/2:Y1=X/2:Y2=-(X
*2+Y1):Y3=-((I+1)*X*2+Y1):Y1=-Y1
  720 LPRINT CHR$(28)+CHR$(37):LPRINT "O":LPRINT "J
1":LPRINT "B1.6":LPRINT "L0"
  730 LPRINT "D";X1;",";Y1;",";X2;",";Y1;",";X2;","
;Y3;",";X1;",";Y3;",";X1;",";Y1
  740 LPRINT "M";X1;",";Y2:LPRINT "L1":LPRINT "D,";
X2;",";Y2
  750 LPRINT "M0,0":LPRINT "S";CS:LPRINT "J2":LPRIN
T "M";X*10;",0":LPRINT "P";M$
  760 FOR J=0 TO I:LPRINT "M";X*5;",";-X*2*(J+1):LP
RINT "J3":LPRINT "P";LEFT$(L$(J),2):LPRINT "J0":LPR
INT "P";MID$(L$(J),3):NEXT J
  770 LPRINT "H":LPRINT CHR$(28)+CHR$(46):LPRINT CH
R$(27);"S1"
  780 RETURN
```

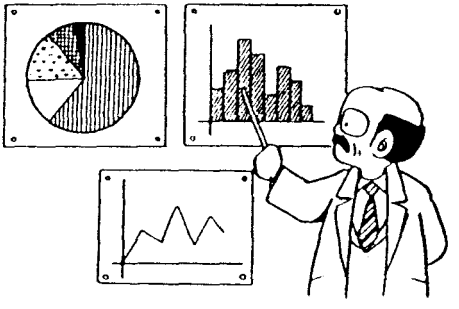
DEBUG

This program permits outputting a calendar on a plotter. If an error occurs with the plotter output, check the routine in lines 700 to 780.

PLOTTER-PRINTER GRAPHIC TOOL

Program Outline

This program takes advantage of the features of CETL to plot simple, practical graphs. Plotted graphs can be prepared from tables (pie chart, band graph, bar graph, jagged-line graph). Since data in CETL can be freely specified to plot graphs, it is possible to modify data and create files speedily and easily.
This program permits the reader to plot graphs and prepare tables using his own data.



Output of Graph

Preparing input data

There are two methods of data input: one is to input data previously stored in CETL, and the other is to input data in response to the request for input made by a BASIC program. By using CETL, it is possible to create graphs and tables speedily and to make corrections easily.
In order to utilize CETL, it is necessary to prepare the data for plotting the desired graph. Data can be stored in up to 10 files.

Starting the program

When the program is started, it displays a comment. If the input data is already stored in CETL, press any key on the keyboard.
The program then displays the prompt "TITLE OF GRAPH:". Input the title of the graph.

Reading data from CETL

After the title is input, the program asks for the CETL file number in which the data is stored. Input the appropriate file number.
The program then asks for the numbers of records and items of the data. Input the numbers of rows and columns of the specified file.
Next, input all the item names (label record).

Selecting graph type

After all the item names are input, the screen displays a menu showning four types of graphs (pie chart, band graph, bar graph, jagged-line graph). Select the desired graph type. The fifth menu item "CETL DATA CHANGE" is used to fetch data from areas other than the specified CETL file area. When this menu item is selected, the program returns to the specification of the file number (Step (3).)

Specifying item or record

After specifying the graph type, specify whether the graph is to be plotted using a record. (Select 1) or using an item (Select 2). (See table below.)

1 Item, N-Record select 2
1 Record, N-Item select 1

YEAR	0~14	15~19	20~29	30~39	40~54	55~59	60~64	65~
1955	2,980	863	1,601	1,123	1,316	321	250	475
'60	2,807	931	1,653	1,356	1,404	364	293	535
'65	2,517	1,085	1,743	1,576	1,554	400	334	618
'70	2,482	906	1,975	1,658	1,802	442	373	733
'75	2,722	795	1,987	1,767	2,137	467	428	887
'80	2,753	823	1,688	2,000	2,356	563	447	1,057
'85	2,574	893	1,599	1,977	2,530	696	540	1,220
'90	2,251	1,001	1,709	1,676	2,776	773	668	1,429
'95	2,141	852	1,887	1,588	2,834	786	743	1,708
2000	2,256	714	1,845	1,697	2,685	865	755	1,994
'05	2,193	734	1,639	1,405	2,477	829	734	2,708
'10	2,191	769	1,587	1,387	2,299	688	606	2,551

Output of Graph

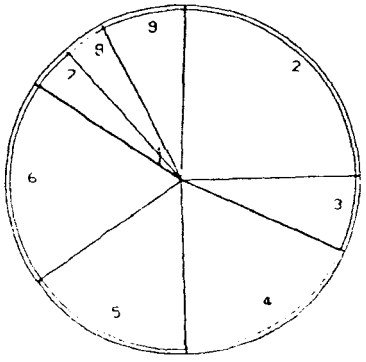
When ❶ is selected, the graph is plotted on a 1-record, N-item basis. First, specify the record number used as the base. Then input the range of each item. Pressing the ↵ key causes the graph to be output on the printer.

When ❷ is selected, the graph is plotted on a 1-item, N-record basis. Specify the item number and input the range of each record, and a graph with the specified item as the base will be output.

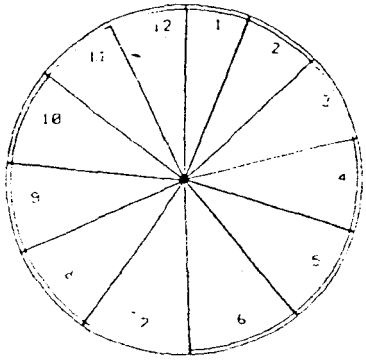
Creating a data table

After the graph is plotted, the data used for plotting the graph is put in a table. When the graph and table are output on the printer, the program returns to the menu screen for selection of a graph type. The program can be terminated by pressing the BREAK key.

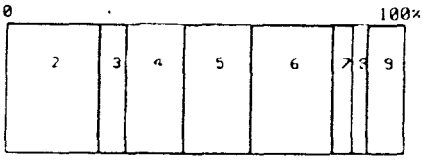
❶ select 1 Pie chart



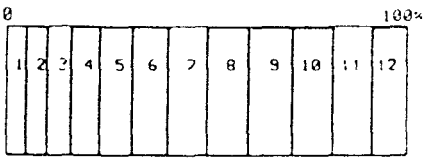
❷ select 2 Pie chart



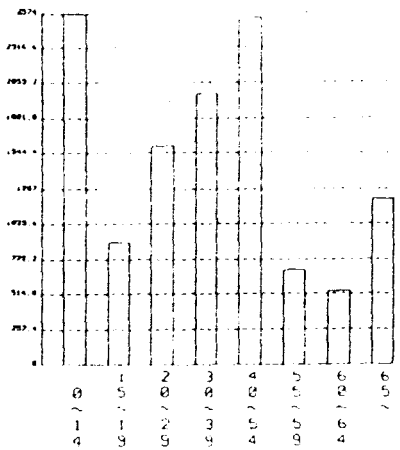
select 1 Band graph



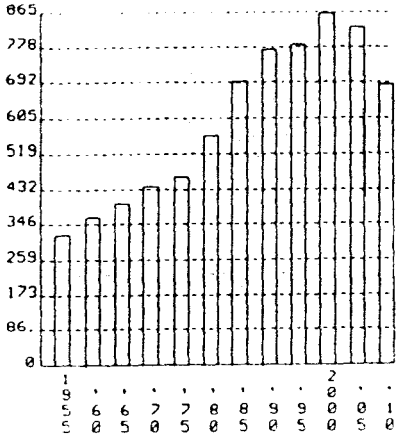
select 2 Band graph



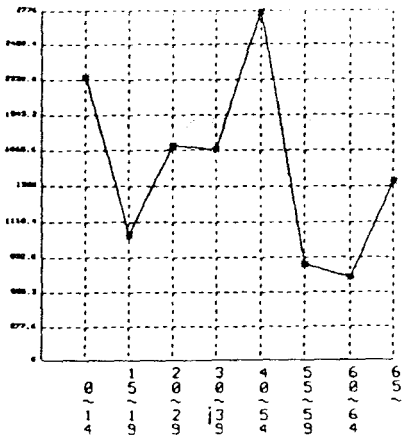
select 1 Bar graph



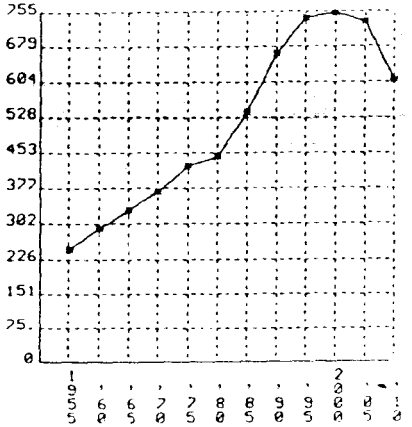
select 2 Bar graph



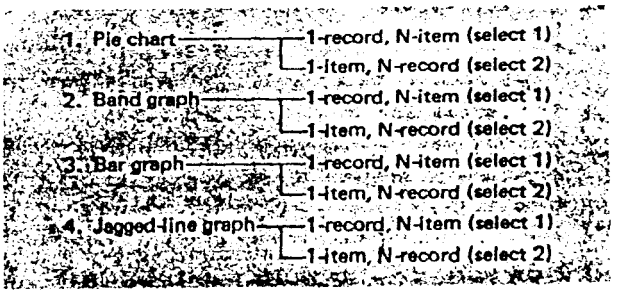
select 1 Jagged-line graph



select 2 Jagged-line graph



Graphs that can be plotted by this program



* Pie charts and band graphs are expressed in percentages.

Input data stored in CETL

YEAR	~14	15~19	20~29	30~39	40~54	55~59	60~64	65~
1955	2,980	863	1,601	1,123	1,316	321	250	475
'60	2,807	931	1,653	1,356	1,404	364	293	535
'65	2,517	1,085	1,743	1,576	1,554	400	334	618
'70	2,482	906	1,975	1,658	1,802	442	373	733
'75	2,722	795	1,987	1,767	2,137	467	428	887
'80	2,753	823	1,688	2,000	2,356	563	447	1,057
'85	2,574	893	1,599	1,977	2,530	696	540	1,220
'90	2,251	1,001	1,709	1,676	2,776	773	668	1,429
'95	2,141	852	1,887	1,588	2,834	786	743	1,708
2000	2,256	714	1,845	1,697	2,685	865	755	1,994
'05	2,193	734	1,639	1,405	2,477	829	734	2,708.
'10	2,191	769	1,587	1,387	2,299	688	606	2,551

```
File Name:PROSPECT

  Number of Rec.:12
  Number of Item:9


Label Record
Item1:Year
   Type(N/S)   :S
   Expression ?
   Format      :&     &

Item2:0~14
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item3:15~19
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item4:20~29
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item5:30~39
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item6:40~54
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item7:55~59
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item8:60~64
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Item9:65~
   Type(N/S)   :N
   Expression ?
   Format      :#,###,###

Data Area
1-1 :1955
1-2 :2980
1-3 :863
1-4 :1601
1-5 :1123
1-6 :1316
1-7 :321
1-8 :250
1-9 :475

2-1 : '60~
2-2 :2807
2-3 :931
2-4 :1653
2-5 :1356
2-6 :1404
2-7 :364
2-8 :293
2-9 :535

3-1 : '65
3-2 :2517
3-3 :1085
3-4 :1743
3-5 :1576
3-6 :1554
3-7 :400
3-8 :334
3-9 :618
```

LIST *4949 Bytes*

```
  100 '
  110 CLEAR 1024:OPTION BASE 1:DIM H(40),SE$(4)
  120 GOSUB 2000
  130 '
  140 CLS:PRINT MI$:PRINT
  150 PRINT "Graph is plotted    using data in CETL
, create data in CETL   or hit key";
  160 IF INKEY$="" THEN 160
  200 '======== Select data
  210 CLS:PRINT MI$
  220 INPUT "Title of graph:",T$
  230 INPUT "CETL file no. ",FO:IF FO<0 OR FO>9 THE
N 220
  240 INPUT "Number of record ",RO:IF RO<1 THEN 240
  250 INPUT "Number of item ",IO:IF IO<1 THEN 250
  260 FOR I=1 TO IO:INPUT "Item name"+STR$(I)+"==>"
,IM$(I):NEXT I
  300 '======== Select graph
  310 CLS:PRINT MI$
  320 FOR I=1 TO 4:PRINT I;"] ";SE$(I):NEXT I
  330 PRINT " 5 ] data modify"
  340 PRINT "select 1~5==>";
  350 SE=VAL(INKEY$):IF SE=0 OR SE>5 THEN 350
  360 IF SE=5 THEN 200
  400 '======== Select records,item
  410 CLS:PRINT "   ** ";SE$(SE);" **"
  420 PRINT "Select record No.       and item No."
  430 PRINT "[1 ] 1RECORD n-ITEM  2 ] 1ITEM n-RECOR
D":PRINT
  440 PRINT "select 1 OR 2==>";
  450 RI=VAL(INKEY$):IF RI=0 OR RI>2 THEN 450 ELSE
PRINT RI
  460 ON RI GOTO 470,490
  470 INPUT "Record No.",RS:IF RS<0 OR RS>RO THEN 4
70
  480 INPUT "Range of Item n,m  ",IS,IE:XS=IS:XE=IE
:IF XS<2 OR XS>IO OR XS>XE THEN 480 ELSE 550
  490 INPUT "Item No. ",IS:IF IS<2 OR IS>IO THEN 49
0
  500 INPUT "Range of record",RS,RE:XS=RS:XE=RE:IF
XS<0 OR XS>RO OR XS>=XE THEN 500
```

```
  550 '======== Print out
  560 CLS:PRINT "     ** ";SE$(SE);" **":PRINT:PRINT
 T$:PRINT "File NO. ";FO
  570 PRINT "record NO.";RS;:IF RI=1 THEN PRINT ELS
E PRINT "~";RE
  580 PRINT "item No. ";IS;:IF RI=1 THEN PRINT "~";
IE ELSE PRINT
  590 ON SE GOTO 600,800,1000,1200
  600 '======== PIE Chart
  610 GOSUB 1600:ANGLE 0:GOSUB 1700
  620 LPRINT "O50,-45":LPRINT "CO,0,40"
  630 A=0:AI=90:J=0
  640 FOR I=XS TO XE:R=H(I)*3.6:AB=A:A=A+R
  650 C1=40:C2=90:C3=AB:GOSUB 700:X1=CX:Y1=CY:C2=90
.4:C3=A:GOSUB 700:X2=CX:Y2=CY:C1=35:C2=90:C3=(A-AB)
/2+AB:GOSUB 700:XM=CX:YM=CY
  660 LPRINT "J";J MOD 4:LPRINT "D";X1;",";Y1;",0,0
,";X2;",";Y2:LPRINT "CO,0,39,";90-A;",";AI
  670 LPRINT "M";XM;",";YM:LPRINT "P";MID$(STR$(I),
2)
  680 AI=90-A:J=J+1
  690 NEXT I:GOTO 1400
  700 '
  710 CX=ROUND(C1*COS(C2-C3),-3):CY=ROUND(C1*SIN(C2
-C3),-3):RETURN
  800 '========
  810 GOSUB 1600:GOSUB 1700:LPRINT "OO,-10"
  820 LPRINT "MO,1.2":LPRINT "PO":LPRINT "M85.2,1.2
":LPRINT "P100%"
  830 LPRINT "DO.8,0,90.8,0,90.8,-30,0.8,-30,0.8,0"
  840 A=0:J=0:C=0.9
  850 FOR I=XS TO XE:S0=1.2:S1=2.4:Z$=MID$(STR$(I),
2):L=LEN(Z$)
  860 X1=A*C+0.8:A=A+H(I):X2=A*C+0.6:IF X2-X1<L*S1
THEN SC=S0 ELSE SC=S1
  870 XM=(X2-X1)/2-L/2*SC+X1:IF L MOD 2<>0 THEN XM=
XM+SC/2
  880 IF I=XE THEN X2=90.8
  890 LPRINT "J";J MOD 4:LPRINT "D";X1;",-0.2,";X2;
",-0.2,";X2;",-29.8,";X1;",-29.8,";X1;",-0.2"
  900 LPRINT "M";XM;",-10":LPRINT "P";Z$
  910 J=J+1
  920 NEXT I:GOTO 1400
 1000 '======== BAR Graph
 1010 GOSUB 1700:GOSUB 1800:C=80/(XE-XS+1)
```

```
 1020 FOR I=XS TO XE:IF RI=1 THEN Z$=IM$(I):K=FL(FO
,RS,I) ELSE Z$=FL(FO,I,1):K=FL(FO,I,IS)
 1030 X2=(I-XS+1)*C+10:X1=X2-C/2:Y=S*K-80:GOSUB 195
0:XM=(X1+X2)/2:YM=-84
 1040 LPRINT "J0":FOR L=1 TO LEN(Z$):LPRINT "M";XM;
",";YM:LPRINT "P";MID$(Z$,L,1):YM=YM-3.6:NEXT L
 1050 LPRINT "J1":LPRINT "D";X1;",-80,";X1;",";Y;",
";X2;",";Y;",";X2;",-80"
 1060 NEXT I:GOTO 1400
 1200 '======== Jagged-LEN
 1210 GOSUB 1700:GOSUB 1800:C=80/(XE-XS+1)
 1220 LPRINT "L1":FOR I=XS TO XE:IF RI=1 THEN Z$=IM
$(I) ELSE Z$=FL(FO,I,1)
 1230 X=(I-XS+1)*C+10:LPRINT "D";X;",0,";X;",-80"
 1240 Y=-84:FOR L=1 TO LEN(Z$):LPRINT "M";X;",";Y:L
PRINT "P";MID$(Z$,L,1):Y=Y-3.2
 1250 NEXT L,I:LPRINT "L0":LPRINT "J1"
 1260 FOR I=XS TO XE:IF RI=1 THEN K=FL(FO,RS,I) ELS
E K=FL(FO,I,IS)
 1270 X=(I-XS+1)*C+10:Y=S*K-80:GOSUB 1950:IF I=XS T
HEN LPRINT "M";X;",";Y ELSE LPRINT "D,";X;",";Y
 1280 LPRINT "N3"
 1290 NEXT I
 1400 '========
 1410 LPRINT "H10":LPRINT "J0":LPRINT CHR$(28);CHR$
(46)
 1420 LPRINT CHR$(27);"S2":LPRINT T$:LPRINT CHR$(27
);"S1"
 1430 IF RI=1 THEN Z1$="ITEM":Z2$=IM$(1)+" "+FL(FO,
RS,1) ELSE Z1$=IM$(1):Z2$=IM$(IS)
 1440 Z2$=LEFT$(Z2$,10):LPRINT "  No. ";Z1$;TAB(27-
LEN(Z2$));Z2$;:IF SE<3 THEN LPRINT " Percentage" EL
SE LPRINT
 1450 LPRINT ML$:J=0
 1460 FOR I=XS TO XE:IF RI=1 THEN Z$=IM$(I):K=FL(FO
,RS,I) ELSE Z$=FL(FO,I,1):K=FL(FO,I,IS)
 1470 IF SE<3 THEN LPRINT CHR$(27);"J";J MOD 4
 1480 LPRINT USING"##### ";I;:LPRINT CHR$(27);"J0":
LPRINT USING"&          & ##,###,###";Z$;K;
 1490 IF SE<3 THEN LPRINT USING" ####.##";H(I):J=J+
1 ELSE LPRINT
 1500 NEXT I:IF SE<3 THEN LPRINT ML$:LPRINT TAB(11)
;"Total";:LPRINT USING "##,###,###";SUM
 1510 LPRINT CHR$(27);"F5"
 1520 GOTO 300
 1600 '======== Caliculate percentage
```

```
 1610 SUM=0:FOR I=XS TO XE:IF RI=1 THEN K=FL(FO,RS,
I) ELSE K=FL(FO,I,IS)
 1620 SUM=SUM+K:NEXT I
 1630 FOR I=XS TO XE:IF RI=1 THEN K=FL(FO,RS,I) ELS
E K=FL(FO,I,IS)
 1640 H(I)=K/SUM*100:NEXT I
 1650 RETURN
 1700 '======== Initialize PRINTER
 1710 LPRINT CHR$(28);CHR$(37):LPRINT "YO":LPRINT "
Z2,6":LPRINT "QO":LPRINT "B2":LPRINT "LO"
 1720 LPRINT "S2":LPRINT "JO":LPRINT "P";T$:LPRINT
"S1"
 1730 RETURN
 1800 '======== Frames for bar,jaggied
 1810 LPRINT "OO,-10":LPRINT "D10,0,10,-80,90,-80"
 1820 MAX=-1E99:MIN=1E99
 1830 FOR I=XS TO XE:IF RI=1 THEN K=FL(FO,RS,I) ELS
E K=FL(FO,I,IS)
 1840 IF MAX<K THEN MAX=K
 1850 IF MIN>K THEN MIN=K
 1860 NEXT I:IF MIN>0 THEN MIN=0
 1870 S=80/(MAX-MIN):Z$=STR$(MAX):IF LEN(Z$)>4 THEN
 LPRINT "SO":SC=1.2 ELSE SC=2.4
 1880 FOR I=MAX TO MIN STEP -(MAX-MIN)/10:Z$=STR$(I
):X=9-SC*LEN(Z$):IF X<0 THEN X=0:Z$=LEFT$(Z$,10/SC)
 1890 Y=S*(I-MIN)-80:GOSUB 1950
 1900 LPRINT "M";X;",";Y:LPRINT "P";Z$:IF I=MIN THE
N 1910 ELSE LPRINT "L1":LPRINT "D10,";Y;",90,";Y:LP
RINT "LO"
 1910 NEXT I:LPRINT "S1"
 1920 RETURN
 1950 '
 1960 IF ABS(Y)<0.1 THEN Y=0
 1970 RETURN
 2000 '======== Setting valiables
 2010 ML$="":FOR I=1 TO 40:ML$=ML$+"-":NEXT I:MI$="
    ** graph  **":SE$(1)="Pie graph"
 2020 RESTORE 2030:FOR I=2 TO 4:READ SE$(I):NEXT I:
RETURN
 2030 DATA Band graph,Bar graph,Jagged graph
```

DEBUG

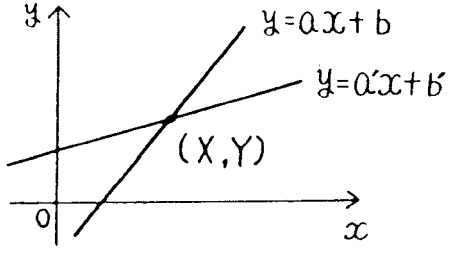
The plotter-printer is put into graphic mode by LPRINT CHR$(28); CHR$(37) in line 1710. It remains in that mode thereafter.

Mathematics and Geometry Programs

POINT OF INTERSECTION OF TWO STRAIGHT LINES

Program Outline

This program obtains the point of intersection of two straight lines on an $x-y$ coordinate plane. Simple as it is, this program can also be used to solve simultaneous equations in two variables.



Data Input

When the program is started, it asks for the two equations to be solved. Thus,

$y = ax + b$

$y = a'x + b'$

Input a, b, a', and b', in that order. The program immediately displays the coordinates of the intersection point of the two straight lines. Thus,

Coordinates of intersection point (x, y)

$x =$

$y =$

If the two straight lines are parallel to each other, the program displays the message "NO INTERSECTION POINT."

If the two straight lines coincide with each other, the program displays "$y = ax + b$", because either of the lines represents the set of intersection points.

Pressing the BREAK key terminates the program execution. To obtain the intersection point of another pair of straight lines, press any other key on the keyboard.

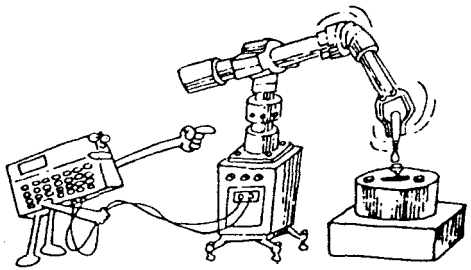
LIST *595 Bytes*

```
100 '*******************************************
110 ' Intersection point of two straight lines
120 '*******************************************
130 CLS
140 PRINT
150 PRINT "y=a x + b"
160 PRINT "y=a'x + b'"
170 LOCATE 8,0:PRINT "Intersection"
180 LOCATE 13,1:PRINT"point"
190 LOCATE 0,5:INPUT "a =";A
200 LOCATE 7,5:INPUT "b =";B
210 LOCATE 0,6:INPUT "a'=";C
220 LOCATE 7,6:INPUT "b'=";D
230 IF A=C AND B=D THEN LOCATE 11,2:PRINT USING "
y=###x+##";A,B:GOTO 280
240 IF A=C THEN LOCATE 11,3:PRINT "Nothing":GOTO
280
250 X=-(B-D)/(A-C)
260 Y=(A*D-C*B)/(A-C)
270 LOCATE 9,3:PRINT USING "(##.#@##.#)";X,",",Y
280 LOCATE 14,5:PRINT "again"
290 LOCATE 14,6:PRINT "anykey";
300 IF INKEY$="" THEN 280
310 GOTO 130
```

HEXADECIMAL CALCULATIONS

Program Outline

Performing the four basic operations, especially multiplication and division, on hexadecimal numbers is a tedious task.
This program permits performing the four basic arithmetic operations on hexadecimal numbers up to five digits.



Data Input

When the program is started, it displays "PLEASE CALCULATE." It is then ready to perform an operation on hexadecimal numbers.
For example, to calculate 7F08 + 00FF, input 7F08 + FF and press the ↵ key. The program then displays the sum, 8007.
Note that when a key which is invalid in hexadecimal notation (e.g., the G key) is pressed, no data is input. Incorrect input data can be corrected by using the cursor key. Note, however, that + and − cannot br corrected. (In this case, the data must be re-input from the beginning.)
Inputting data exceeding five digits does not cause an error to be indicated, but the result obtained is incorrect.
Subtractions, multiplications and divisions can be performed in the same manner as described above. Note, however, that the routine to perform divisions on negative hexadecimal numbers has been omitted because such divisions are seldom performed.
When performing the next operation, press the SPACE key after completing the preceding operation. To terminate the program, press the BREAK key.
In divisions, digits below the decimal point are truncated.

■ Examples of calculation:

```
Please Calculate
7F08+FF=8007

Please Calculate
7F00-184F=66B1

Please Calculate
FFFFF*FFFFF=FFFFE000
01

Please Calculate
FFFF2/74F=230
```

LIST

1236 Bytes

```
   10 '??????????????????????????????????????????????
??????
   20 '
   30 ' From basic arithmetics eperations in Hexade
cinal
   40 '                                          by IY
O
   50 '??????????????????????????????????????????????
??????
   60 CLEAR:DIM A$(10),A(10),B$(20):DEFDBL T
   70 FOR J=0 TO 9
   80 B$(J)=RIGHT$(STR$(J),1)
   90 NEXT J
  100 FOR J=65 TO 70
  110 B$(J-55)=CHR$(J)
  120 NEXT J
  130 CLS:PRINT"Please Calculate"
  140 K$=INKEY$:IF K$="" THEN 140
  150 IF K$=CHR$(29) THEN A$(I)=MID$(A$(I),1,LEN(A$
I))-1):PRINT CHR$(29);CHR$(32);CHR$(29);:GOTO 140
  160 IF K$=CHR$(13) THEN 220
  170 IF K$="+" OR K$="-" OR K$="*" OR K$="/" THEN
=I+1:A$(I)=K$:I=I+1:GOTO 210
  180 Z=ASC(K$)
  190 IF Z<48 OR Z>57 AND Z<65 OR Z>70 THEN GOTO 14

  200 A$(I)=A$(I)+K$
  210 PRINT K$;:GOTO 140
  220 FOR N=0 TO I STEP 2
  230 GOSUB 430
  240 NEXT N
  250 T=A(0)
  260 FOR J=2 TO I STEP 2
  270 IF A$(J-1)="+" THEN T=T+A(J):GOTO 310
  280 IF A$(J-1)="-" THEN T=T-A(J):GOTO 310
  290 IF A$(J-1)="*" THEN T=T*A(J):GOTO 310
  300 T=T/A(J)
  310 NEXTJ
  320 IF T<0 THEN T=ABS(T):X$="-":GOTO 340
  330 X$=""
  340 Y=FRAC(T/16)*16
```

```
360 FOR K=1 TO N
370 IF K=I THEN 420
380 C=A(K,I)
390 FOR J=1 TO N:A(K,J)=A(K,J)-C*A(I,J)
400 NEXT J
410 B(K)=B(K)-C*B(I)
420 NEXT K
430 NEXT I
440 CLS:LOCATE0,0:PRINT"****  Answer  ****"
450 FOR I=1TO N
460 LOCATE1,I:PRINT R$(I)+"=";CSNG(B(I))
470 NEXT I
480 Z$=INKEY$:IF Z$="" THEN 480 ELSE IF Z$="E" OR
Z$="e" THEN END
490 GOTO20
500 END
510 CLS:LOCATE0,2:PRINT"  Answer cannot
be obtained        from this data!!"
520 GOTO 480
```

DEBUG

Although this program can handle only a maximum of six variables due to space limitations on the display screen, it can be modified such that it can solve simultaneous equations in as many variables as the memory capacity permits. This can be done as follows:

(1) Declare as many A (I, J) and B (I) arrays as required using the DIM command.
(2) Change the variables associated with the screen display in the following lines:
- Lines 30 and 40 (create character expressions used for display)
- Line 110 (display general expressions)
- Lines 120 to 160 (input coefficients and constants)
- Lines 440 to 460 (display solutions)

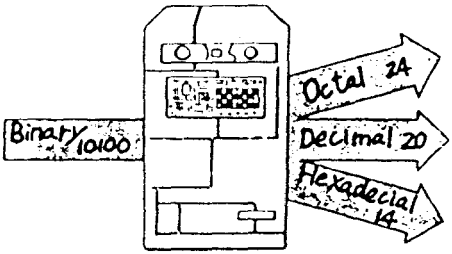
In this case, change the character variable arrays accordingly.

When the number of variables involved in simultaneous equations is large, it will not be possible to display all of the equations at one time on the display screen of the FP–200. In this case, appropriate measures can be taken to display them in several steps.

CONVERSION BETWEEN BINARY, OCTAL, DECIMAL AND HEXADECIMAL

Program Outline

This program permits conversion between binary, octal, decimal, and hexadecimal numbers.
Binary notation has been increasingly used in machine languages, etc. Converting binary numbers into decimal, octal, or hexadecimal numbers is not so simple. This program makes such conversions and easy task.



Data Input

When the program is stared, it displays and explanatory message.
Press any key on the keyboard and the program prompts for data input. Thus,

B) Binary
O) Octal
D) Decimal
H) Hexadecimal

Specify B, O, D, or H for the data to be input.
The program then asks for the value of the input data. Input the number. Pressing the ↵ key causes the input number and converted values to be displayed on the screen.
For example, when you select B (binary), the program displays 'Binary to D/O/H.' Input a binary number and the program displays the binary number, together with its octal, decimal, and hexadecimal equivalents.
Pressing the ↵ key causes a return to the beginning of the program.
Any positive integer from 0 to 65535 can be converted by this program. Hence, a valid number should not exceed the number of digits represented by the number of X's in the program. Inputting data beyond the applicable range causes a return to the beginning of the program (prompt for data input). Using an invalid number (e.g., anything except 0 or 1 in binary numbers, 8 and 9 in octal numbers, anything except 0–9, A–F in hexadecimal numbers) also causes a return to the beginning of the program.
To terminate the program, press the BREAK key.

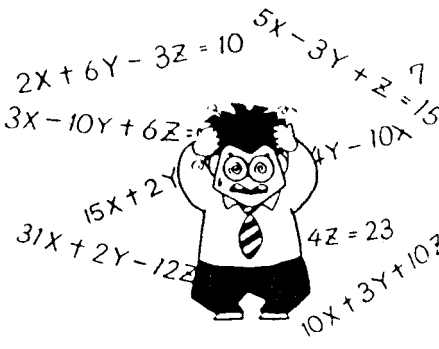
Examples of program execution (screen display)

	* Binary to O/D/H *
Convert a binary number to octal, decimal, and hexadecimal numbers.	Binary data (XXXXXXXXXXXXXXXX)
Input 1011 (binary number)	1011
Display 1011 (binary number)	Bin: 1011
Display 13 (octal number)	Oct: 13
Display 11 (decimal number)	Dec: 11
Display B (hexadecimal notation)	Hex: B

SIMULTANEOUS EQUATIONS OF THE FIRST DEGREE

Program Outline

Of various mathematical operations, simultaneous linear equations in several variables are used for a variety of applications. Although the concepts and method of calculation of simultaneous linear equations in several variables are simple and plain, solving those equations with many variables is not so simple.
This program permits the FP-200 to solve simultaneous linear equations in several variables.



Data Input

When the program is started, it asks for the number of variables in the simultaneou linear equations. Since this program can solve simultaneous linear equations with up t six variables, the maximum number of variables that can be specified is 6.
In order to solve simultaneous equations in n variables, it is necessary that there be different equations. Input the coefficients and constants of the n equations. Fo example, here are simultaneous equations in four variables.

$$\begin{cases} A_1x_1 + B_1x_2 + C_1x_3 + D_1x_4 = E_1 \cdots\cdots ❶ \\ A_2x_1 + B_2x_2 + C_2x_3 + D_2x_4 = E_2 \cdots\cdots ❷ \\ A_3x_1 + B_3x_2 + C_3x_3 + D_3x_4 = E_3 \cdots\cdots ❸ \\ A_4x_1 + B_4x_2 + C_4x_3 + D_4x_4 = E_4 \cdots\cdots ❹ \end{cases}$$

When equation ❶ is displayed, input A1, B1, C1, D1, and E1, in that order. The pro gram then displays the prompt "OK?".
To correct the input data, press the N key. When there is no need to correct the inpu data, press any key except the N key.
When you press the N key, the program waits for input of data from the beginnin (A1). When you press any key except the N key, the program waits for input of th coefficients and constant of equation ❷.
When all the coefficients and constants have been input, the program immediatel starts the computation and displays the results.
Pressing the E key terminates program execution. To perform the calculation of anothe set of simultaneous equations, press any key except the E key.

■ **Exercise:**

$$\begin{cases} 2x_1 + 3x_2 - 4x_3 + x_4 = 11 \cdots\cdots ❶ \\ -x_1 + 4x_2 + x_3 - 2x_4 = 14 \cdots\cdots ❷ \\ 7x_1 - 2x_2 + 9x_3 = 8 \cdots\cdots ❸ \\ -3x_1 + x_2 - 7x_3 - x_4 = 5 \cdots\cdots ❹ \end{cases}$$

Input of the number of variables

```
*** Simultaneous linear equations ***
Number of variables = ? 4
```

Input of coefficients and constants

```
A 1 x 1 + B 1 x 2 + C 1 x 3 + D 1 x 4 =
E 1
A 1 = 2                    E 1 = 11
B 1 = 3
C 1 = -4                   ok
D 1 = -1                   yes anykey
                           no : (N)
```

Output of solutions

```
**** Answer ****
x 1 =  3
x 2 =  2
x 3 = -1
x 4 = -5
```

LIST

1260 Bytes

```
 10 CLEAR:DEFDBLA-C
 20 CLS:PRINT"** Simultaneous "
 30 PRINT" linear equatious **"
 40 INPUT"Number of variables  =";N:IF N>6 THEN 4
0
 50 FOR I=1 TO N+1:P$(I)=CHR$(I+64):Q$(I)=CHR$(I+
48):R$(I)="X"+Q$(I):NEXT I
 60 FOR I=1 TO N:S$=""
 70 FOR J=1 TO N
 80 S$=S$+P$(J)+Q$(I)+R$(J)+"+"
 90 NEXT J
100 T$=LEFT$(S$,5*N-1)+"="+P$(N+1)+Q$(I)
110 CLS
120 LOCATE 0,0:PRINTT$
130 FOR J=1 TO N
140 U$=P$(J)+Q$(I)+"="
150 LOCATE 0,J+1:PRINT U$;:INPUT"",A(I,J)
160 NEXT J
170 LOCATE10,2:PRINT P$(N+1)+Q$(I)+"=";:INPUT"",B
(I)
180 LOCATE11,4:PRINT"ok ?"
190 LOCATE10,5:PRINT"yes:anykey"
200 LOCATE10,6:PRINT"no :[N]"
210 Z$=INKEY$:IF Z$=""THEN 210 ELSE IFZ$="N" OR Z
$="n" THEN 70
220 NEXT I
230 CLS:PRINT"Being computed"
240 '
250 FOR I=1 TO N
260 J=I
270 IF A(I,J)<>0 THEN 320
280 J=J+1:IF J>N THEN 510
290 IF A(I,J)=0 THEN 270
300 FOR K=1 TO N:C=A(I,K):A(I,K)=A(J,K):A(J,K)=C:
NEXT K
310 C=B(I):B(I)=B(J):B(J)=C
320 C=A(I,I)
330 IF C=0 THEN 510
340 FOR J=1 TO N:A(I,J)=A(I,J)/C:NEXT J
350 B(I)=B(I)/C
```

LIST *2210 Bytes*

```
  100 '
  110 CLEAR 1024
  120 '
  130 CLS:PRINT
  140 PRINT "Conversion between  binary,octal,decim
aland hexadecimal num-bers.
  150 IF INKEY$="" THEN 150
  200 '
  210 CLS:PRINT
  220 PRINT "INPUT B] Binary"
  230 PRINT "      O] Octal"
  240 PRINT "      D] Decimal"
  250 PRINT "      H] Hexadecimal"
  260 PRINT "select B/O/D/H"
  270 Z$=INKEY$:IF Z$="B" OR Z$="b" THEN 400
  280 IF Z$="O" OR Z$="o" THEN 500
  290 IF Z$="D" OR Z$="d" THEN 600
  300 IF Z$="H" OR Z$="h" THEN 700
  310 GOTO 270
  400 '========== binary input
  410 CLS:PRINT "* Binary to O/D/H *"
  420 PRINT "Binary data"
  430 PRINT " (XXXXXXXXXXXXXXX)"
  440 INPUT "  ",B$:X$=B$
  450 A=2:GOSUB 1000:IF ER=1 THEN 420
  460 A=8:GOSUB 1200:O$=X$
  470 GOSUB 1300:H$=X$
  480 CLS:PRINT "Bin:";B$:PRINT "Oct:";O$:PRINT "De
c:";D$:PRINT "Hex:";H$
  490 GOTO 800
  500 '========== Octal input
  510 CLS:PRINT "* Octal to B/D/H *"
  520 PRINT "Octal data (XXXXXX)"
  530 INPUT "             ",O$:X$=O$
  540 A=8:GOSUB 1000:IF ER=1 THEN 520
  550 A=2:GOSUB 1200:B$=X$
  560 GOSUB 1300:H$=X$
  570 CLS:PRINT "Oct:";O$:PRINT "Bin:";B$
  580 PRINT "Dec:";D$:PRINT "Hex:";H$
  590 GOTO 800
```

```
  600 '========== Decimal input
  610 CLS:PRINT "* Decimal to B/O/H *"
  620 PRINT "Decimal data(XXXXX)"
  630 INPUT "                ",D$
  640 D=VAL(D$):IF D>65535 THEN 620
  650 A=2:GOSUB 1200:B$=X$
  660 A=8:GOSUB 1200:O$=X$
  670 GOSUB 1300:H$=X$
  680 CLS:PRINT "Dec:";D$:PRINT "Bin:";B$:PRINT "Oc
t:";O$:PRINT "Hex:";H$
  690 GOTO 800
  700 '========== Hexadecimal input
  710 CLS:PRINT "  * Hex to B/O/D *"
  720 PRINT "Hex data (XXXX)"
  730 INPUT "            ",H$:X$=H$
  740 GOSUB 1100:IF ER=1 THEN 720
  750 A=2:GOSUB 1200:B$=X$
  760 A=8:GOSUB 1200:O$=X$
  770 CLS:PRINT "Hex:";H$:PRINT "Bin:";B$
  780 PRINT "Oct:";O$:PRINT "Dec:";D$
  800 '==========
  810 IF INKEY$="" THEN 810 ELSE GOTO 200
 1000 '========== 2/8 TO 10
 1010 X=0:Y=1:ER=0
 1020 FOR I=1 TO LEN(X$)
 1030 W=ASC(RIGHT$(X$,I))-48
 1040 IF W<0 OR W>A-1 THEN ER=1:GOTO 1080
 1050 X=X+W*Y:Y=Y*A
 1060 NEXT I:IF X>65535 THEN ER=1:GOTO 1080
 1070 D$=MID$(STR$(X),2):D=X
 1080 RETURN
 1100 '========== 16 TO 10
 1110 X=0:Y=1:ER=0
 1120 FOR I=1 TO LEN(X$)
 1130 W=ASC(RIGHT$(X$,I))-48
 1140 IF W<0 OR W>=10 AND W<=16 OR W>22 THEN ER=1:G
OTO 1190
 1150 IF W>10 THEN W=W-7
 1160 X=X+W*Y:Y=Y*16
 1170 NEXT I:IF X>65535 THEN ER=1:GOTO 1190
 1180 D$=MID$(STR$(X),2):D=X
 1190 RETURN
 1200 '========== 10 TO 2/8
 1210 X=D:X$=""
 1220 Y=INT(X/A):W=X-Y*A:X=Y
```

```
1230 X$=CHR$(W+48)+X$
1240 IF X<>0 THEN 1220
1250 RETURN
1300 '========== 10 TO 16
1310 X=D:X$=""
1320 Y=INT(X/16):W=X-Y*16:X=Y
1330 IF W>9 THEN W=W+7
1340 X$=CHR$(W+48)+X$
1350 IF X<>0 THEN 1320
1360 RETURN
```

```
350 C$=B$(Y)+C$
360 T=T-Y
370 T=INT(T/16)
380 IF T>15 THEN 340
390 C$=B$(T)+C$
400 PRINT "=";X$;C$
410 IF " "=INKEY$ THEN 60
420 GOTO 410
430 L=LEN(A$(N))
440 FOR M=1 TO L
450 D$=MID$(A$(N),M,1)
460 IF D$>="0" AND D$=<"9" THEN V=VAL(D$):GOTO 48
0
470 V=ASC(D$)-55
480 A(N)=A(N)+V*16^(L-M)
490 NEXT M
500 RETURN
```

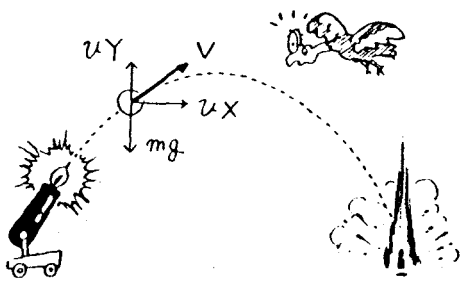
DEBUG

In this program, all input data is first converted into decimal data for computation and the result obtained is then recoverted into the appropriate hexadecimal data for display. If an error occurs during computation, check lines 270 to 300.

PARABOLA

Program Outline

Parabolic motion is a common motion in physics. It represents the path followed by an object thrown up at an angle. This program draws parabolas using the dot-graphics capability of the FP-200.



Data Input

When the program is started, it displays coordinates, waiting for input of data. Input the following data:

g: Gravitational acceleration (normally 9.8) (m/s^2)
h: Height from which object is thrown up (m)
Vo: Initial velocity (m/s)
θ: Angle at which object is thrown (degrees, measured with respect to the horizontal plane)

If you wish to divide the initial velocity into x and y components, input 0 for Vo. By so doing, it is possible to input the initial velocity by Vx and Vy.

When the data is input, the program puts graduations on the x–y coordinate axes and plots a parabola. Displayed to the right of the parabola are the current values of the x and y coordinates, time (t), and scale indicating the actual distance (m) represented by one graduation. When the value of x is negative, $-x$ is displayed on the graph.

The program temporarily stops when the dot reaches the highest point. Press any key on the keyboard to continue execution.

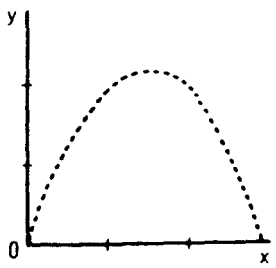
When the object falls to the ground ($y=0$), program execution is terminated, displaying the final velocity at the bottom right of the screen. Press the keys X, Y, V, and 0, and the following data are displayed:

X: x–component of final velocity (m/s)
Y: y–component of final velocity (m/s)
V: Final velocity (same as initial velocity) (m/s)
0: Angle as measured from the ground (0 degrees indicates the horizon to the right)

Pressing any other key terminates the program.

The minimum scale value is 1. Hence, specifying an extremely small scale value (values of x and y become fractions with two or more digits below the decimal point) may prevent the program from plotting the parabola. Note also that very large values of x and y cause the parabola to become somewhat deformed.

Screen output



g =9.8
h =50
Vo=500
X =16415.82
Y =0.00
T =95.99
V =500.98
O =70

LIST

1583 Bytes

```
 10 '***********
 20 '
 30 ' Parabolas
 40 '
 50 '***********
 60 CLEAR
 70 ANGLE 0
 80 INIT(0,0),1,1
 90 CLS
100 DRAW (8,0)-(8,55)
110 DRAW (8,55)-(63,55)
120 DRAW (68,0)-(68,63)
130 LOCATE 0,7:PRINT "0";
140 LOCATE 0,1:PRINT "y"
150 LOCATE 6,7:PRINT "x";
160 LOCATE 9,0:INPUT "g=",G
170 LOCATE 9,1:INPUT "h=",H
180 LOCATE 9,2:INPUT "Vo=",V0
190 IF V0=0 THEN 240
200 LOCATE 9,3:INPUT "0=",TH
210 VX=COS(TH)*V0
220 VY=SIN(TH)*V0
230 GOTO 270
240 LOCATE 9,2:PRINT "          "
250 LOCATE 9,2:INPUT "Vx=",VX
260 LOCATE 9,3:INPUT "Vy=",VY
270 SG=1
280 IF VX<0 THEN VX=-VX:SG=-1:LOCATE 6,7:PRINT "-
x";
```

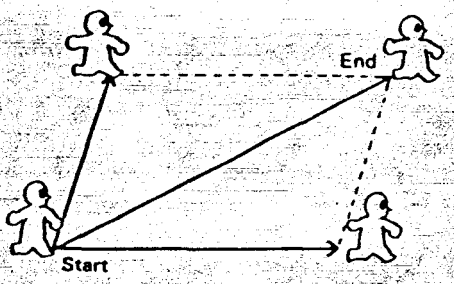
```
 290 TT=VY/G
 300 TX=VX*TT
 310 TY=H+VY*TT-G*TT^2/2
 320 LT=TT+SQR(2*TY/G)
 330 LX=VX*LT
 340 IF VY<0 THEN TY=H
 350 SY=50/TY
 360 SCX=1:SCY=1
 370 IF TY>LX THEN SCX=LX/TY:M=TY ELSE SCY=TY/LX:M
=LX
 380 M=M/5:M1=1:N=2
 390 IF M>10 THEN M1=M1*10:M=M/10:GOTO 390
 400 FOR I=0 TO 3
 410  N=N+2*I-1
 420  IF M>N THEN 440
 430  I=3
 440 NEXT I
 450 N=N*M1:M=M*M1
 460 FOR I=0 TO 60 STEP 50/(M*5)*N
 470  IF I>55 OR  I=0 THEN 490
 480  DRAW (7,55-I),(8+I,56)
 490 NEXT I
 500 IF VY<=0 THEN J=1:SX=50/LX:GOTO 670
 510 SX=TT/LT
 520 LOCATE 9,7:PRINT USING "SCALE#####";N;
 530 FOR I=1 TO 50*SX
 540  X=TX*I/(50*SX)
 550  T=TT*I/(50*SX)
 560  Y=(H+VY*T-G*T^2/2)
 570  GOSUB 780
 580 NEXT I
 590 T=TT
 600 X=TX
 610 Y=TY
 620 GOSUB 780
 630 J=I
 640 LOCATE 9,7:PRINT "HIT ANYKEY";
 650 IF INKEY$="" THEN 650
 660 LOCATE 9,7:PRINT "           ";
 670 LOCATE 9,7:PRINT USING "SCALE#####";N;
 680 FOR I=J TO 50
 690  X=LX*I/50
 700  T=LT*I/50
 710  Y=(H+VY*T-G*T^2/2)
 720  GOSUB 780
```

```
730 NEXT I
740 V=SQR(VX^2+(-VY+LT*G)^2)
750 LOCATE 9,7:PRINT USING "V=######.##";V;
760 IF INKEY$="" THEN 760
770 END
780 LOCATE 9,4
790 PRINT USING "X=#####.##";X*SG
800 LOCATE 9,5
810 PRINT USING "Y=#####.##";Y
820 LOCATE 9,6
830 PRINT USING "T=#####.##";T
840 DRAW (8+I*SCX,55-Y*SY*SCY)
850 RETURN
```

VECTOR ADDITION AND SUBTRACTION

Program Outline

Vector additions and subtractions are easy to perform. However, it is not so easy to see the quantity and direction of vectors thus calculated. This program takes advantage of the dot-graphics capability of the FP-200 to represent vectors in the form of graphs.



Operations

When the program is started, it displays the message shown in ❶, waiting for data input. The scale in this message indicates the amount represented by a single dot. It determines the ranges of X and Y that can be input. The message shows that the current scale is 1.0/d (dot), indicating that both X and Y have a range of −31.0 to 31.0.

The program then asks for a new scale. Determine the new scale based on the magnitudes of vectors to be input. (In this case, there is no need to consider the magnitude of the resulting vector.) When you input a scale of, say, 0.5, the program again displays the scale for confirmation as shown in ❷. Press any key on the keyboard.

Now, the screen appears as shown in ❸. Drawn to the right of the scale is an X–Y plane, and the intersecting point in the center is the origin. Input the *x* and *y* component of the position vector. When the data is input, the program asks whether an addition or a subtraction is to be performed. Key in a '−' for subtraction. Press the ↵ key for addition.

The program then draws the input vector as shown in ❹. In the case of subtraction, a '−' is placed in front of the first parenthesis. Up to 11 vectors can be input for calculation. Note that when more than five vectors are input, the preceding vectors disappear sequentially to prevent the screen display from overflowing. (The result of the calculation is unaffected.)

After inputting all the vectors, input 0 for X and Y. The program then displays the result, which blinks. If the result exceeds the range that can be displayed on the current scale, the scale is automatically adjusted such that the display can be continued. Pressing any key on the keyboard causes a return to the beginning of the program.

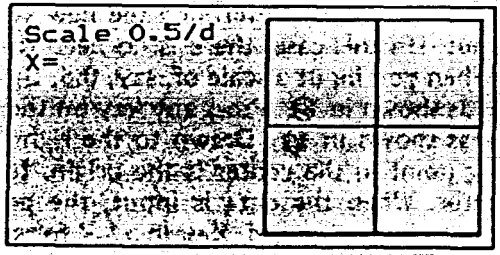
Display ❶

```
Scale 1.0/d now
X -31..0..31
Y -31..0..31
Scale=
```

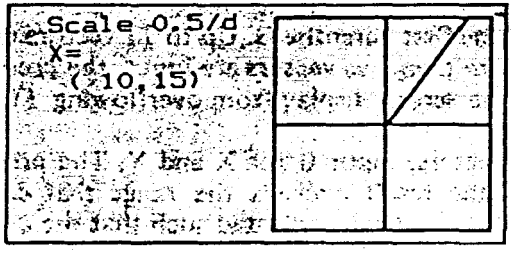
❷

```
X -31..0..31
Y -31..0..31
Scale= 0.5
Scale 0.5/d
X -15.5 ..0.. 15.5
Y -15.5 ..0.. 15.5
Hit any key
```

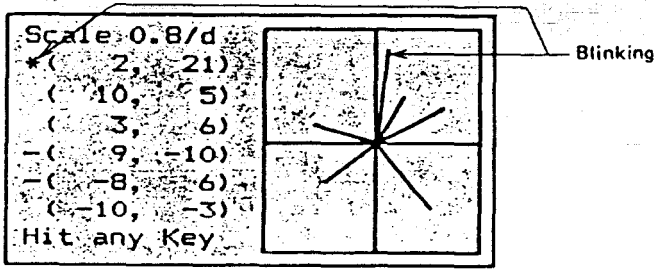
❸



❹



❺



LIST *1267 Bytes*

```
 10 CLEAR:CLS:S=1
 20 INIT(127,31),1,1
 30 GOSUB 480:LOCATE 12,0:PRINT"now"
 40 PRINT "X -31..0..31","Y -31..0..31"
 50 INPUT"Scale=",S:U=S:IF S<.1 THEN 50
 60 GOSUB 480
 70 PRINT "X ";-31*S;"..0..";31*S,"Y ";-31*S;"..0
..";31*S
 80 PRINT "Hit any key"
 90 IF INKEY$="" THEN 90
100 CLS
110 DRAW(-32,0)-(31,0)
120 QUAD(-32,-31)-(32,31)
130 DRAW(0,-31)-(0,31)
140 INIT(127,31),1/S,1/S
150 LOCATE0,0:GOSUB 480
160 LOCATE0,1:INPUT"X=",X:LOCATE2,1:PRINT "      "
170 IF X>31*S OR X<-31*S THEN 160
180 LOCATE0,1:INPUT"Y=",Y:Y=-Y:LOCATE2,1:PRINT "
    "
190 IF -Y>31*S OR -Y<-31*S THEN 180
200 IF X=0  AND Y=0 THEN 330
210 LOCATE0,1:INPUT"+ OR - ",F$:LOCATE0,1:PRINT"
      "
220 IF F$="-" THEN F=-1 ELSE F=1
230 DRAW(0,0)-(X,Y)
240 XT=XT+X*F
250 YT=YT+Y*F
260 X(I)=X
270 Y(I)=Y
280 IF F$="-" THEN LOCATE0,II+2:PRINT F$
290 LOCATE1,II+2:PRINT USING"(###@###)";X,",",-Y;
300 I=I+1:II=II+1
310 IF II=5 THEN II=0
320 GOTO 160
330 IF ABS(XT)>ABS(YT) THEN T=ABS(XT) ELSE T=ABS(
YT)
340 IF T>31*S THEN S=S+.2:IF (S*10)MOD2=1 THEN S=
S+.1:GOTO 340
350 FOR H=0 TO I
```

```
360 DRAWC(0,0)-(X(H),Y(H))
370 NEXT H
380 INIT(127,31),1/S,1/S
390 LOCATE 0,0:GOSUB 480
400 FOR H=0 TO I
410 DRAW(0,0)-(X(H),Y(H))
420 NEXT H
430 LOCATE1,1:PRINT USING"(###@###)";XT,",",-YT
440 LOCATE0,7:PRINT "Hit any key";
450 DRAW(0,0)-(XT,YT):LOCATE0,1:PRINT "*"
460 IF INKEY$="" THEN DRAWC(0,0)-(XT,YT):LOCATE0,
1:PRINT " ":GOTO 450
470 GOTO 10
480 IF S>=10 THEN 490 ELSE PRINT USING"Scale #.#/
d";S:RETURN
490 PRINT USING"Scale ###/d";S:RETURN
```

DEBUG

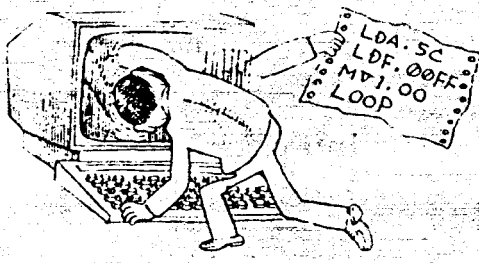
When moving from this program to another program using PROG, be sure to execute 'INIT (0,0), 1,1' beforehand. Otherwise, an error may occur with a program using graphics capability. Note also that excessively large scale or component values may deform the display on the screen.

System Development Programs

DISASSEMBLER

Program Outline

This is a disassembler for the 80C85, which is the FP-200 CPU. The functions of a disassembler are indispensable for understanding programs machine language or for checking the internal operation of the FP-200.
This program reads the contents of memory addresses specified by the PEEK command and displays the associated assembly language.

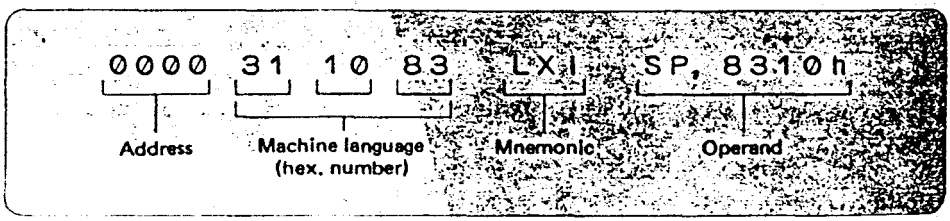


Data Input

When the program is started, it asks whether or not a printer is used. If you use a printer is used. If you use a printer, press the Y key. If not, press the N key.
Then input the starting address and the ending address of the area to be disassembled, in hexadecimal. If the ending address is smaller than the starting address or if a non-existent address is input, the program prompts for re-input of data.
When a printer is used, the program continuously outputs all the addresses from the starting address to the ending address, the machine language (hexadecimal notation) written in those addresses, and the disassembled contents, on the printer immediately after the addresses are input.
When no printer is used, the program displays the contents at a rate of three items per screen. To display the next three items, press any key on the keyboard. Repeat this operation to display all the contents to the ending address.
At the end of disassembly, the program again prompts for input of the starting address and the ending address of another area to be disassembled. To terminate the program, press the BREAK key.
In this disassembler, when a printer is used, one line of blank space is provided after each JMP or RET command. This is to facilitate identifying the point where the program flow is changed.
If the address does not contain any command, the result of disassembling is output as follows:

address· memoric operand

Output format



Example of printout when a printer is used

```
0010  7C          MOV  A,H
0011  BA          CMP  D
0012  C0          RNZ
0013  7D          MOV  A,L
0014  BB          CMP  E
0015  C9          RET

0016  00          NOP
0017  C3 7E 23     JMP  237Eh
```

Example of output when printer is not used

```
0000 31 10 83
      LXI  SP,8310h
0003 3E CA
      MVI  A,CAh
0005 C3 81 00
      JMP  0081h
```

ress any key on the keyboard.

```
0008 1A
      LDAX D
0009 FE 20
      CPI  20h
000B C0
      INX  D
```

LIST

2935 Bytes

```
  1 '*********************************
  2 '
  3 '  Dis-assembler  8085  V.1
  4 '
  5 '       copyright 1983 A.Kakizono
  6 '
  7 '          1983/ 6/ 9
  8 '
  9 '*********************************
100 '
110 CLEAR 1023
120 DIM N1$(63),N2$(7),N3$(63),REG$(7)
130 GOSUB 1000
140 '
150 CLS:PRINT "* Dis-assembler V.1*"
160 INPUT "Printer(Y/N) :",X$:IF X$="Y" OR X$="y"
 THEN PSW=1 ELSE PSW=0
200 '
210 INPUT "top addr(HEX):",Z$:GOSUB 600:TA=Z:IF E
R=1 THEN 210
220 INPUT "end addr(HEX):",Z$:GOSUB 600:EA=Z:IF E
R=1 THEN 220
230 IF TA>=EA THEN 200 ELSE A=TA:J=0
300 '
310 Z=A:L=4:GOSUB 700:AD$=Z$:M=PEEK(A):Z=M:L=2:GO
SUB 700:MC$=Z$:OP$=""
320 IF M<=63 THEN NM$=N1$(M):GOTO 400
330 IF M=118 THEN NM$="HLT":GOTO 500
340 IF M>=64 AND M<=127 THEN NM$="MOV  ":OP$=REG$
((M AND 56)/8)+","+REG$(M MOD 8):GOTO 500
350 IF M>=128 AND M<=191 THEN NM$=N2$((M AND 120)
/8)+"  ":OP$=REG$(M MOD 8):GOTO 500
360 NM$=N3$(M-192)
400 '
410 IF NM$="" THEN NM$=".    ":OP$="'"+CHR$(M):GO
TO 500
420 N$=LEFT$(NM$,1):IF N$<"2" OR N$>"3" THEN 500
430 NM$=MID$(NM$,2):IF RIGHT$(NM$,1)="/" THEN NM$
=LEFT$(NM$,LEN(NM$)-1)+","
440 IF N$="2" THEN A=A+1:Z=PEEK(A):L=2:GOSUB 700:
OP$=Z$+"h":MC$=MC$+" "+Z$:GOTO 500
```

```
  450 Z=PEEK(A+1):L=2:GOSUB 700:Z1$=Z$:Z=PEEK(A+2):
GOSUB 700:Z2$=Z$:OP$=Z2$+Z1$+"h":MC$=MC$+" "+Z1$+"
"+Z2$:A=A+2
  500 '
  510 Z$=AD$+"  "+LEFT$(MC$+"          ",10)+NM$+OP$
  520 IF PSW=0 THEN PRINT LEFT$(Z$,16):PRINT TAB(6)
;MID$(Z$,17) ELSE LPRINT Z$
  530 IF (LEFT$(NM$,3)="JMP" OR LEFT$(NM$,3)="RET")
AND PSW=1 THEN LPRINT:J=J+1
  540 J=J+1:IF PSW=1 THEN 570
  550 IF J<3 THEN 570 ELSE J=0
  560 IF INKEY$="" THEN 560 ELSE CLS
  570 A=A+1:IF A<=EA THEN 300
  580 END
  600 ' hex$(z$) to dec(z)
  610 Z=0:Y=1:ER=0:FOR I=1 TO LEN(Z$):W=ASC(RIGHT$(
Z$,I))-48
  620 IF W<0 OR W>=10 AND W<=16 OR W>22 THEN ER=1:G
OTO 660
  630 IF W>16 THEN W=W-7
  640 Z=Z+W*Y:Y=Y*16
  650 NEXT I:Z$=RIGHT$("000"+Z$,4)
  660 RETURN
  700 ' dec(z) to hex$(z$)
  710 Z$=""
  720 Y=INT(Z/16):W=Z-Y*16:Z=Y:IF W>9 THEN W=W+7
  730 Z$=CHR$(W+48)+Z$:IF Z<>0 THEN 720
  740 Z$=RIGHT$("000"+Z$,L)
  750 RETURN
 1000 '
 1010 RESTORE 1100:FOR I=0 TO 63:READ N1$(I):NEXT I
 1020 FOR I=0 TO 7:READ N2$(I):NEXT I
 1030 FOR I=0 TO 63:READ N3$(I):NEXT I
 1040 FOR I=0 TO 7:READ REG$(I):NEXT I
 1050 RETURN
 1100 '
 1110 DATA NOP,3LXI  B/,STAX,INX  B,INR  B,DCR  B,2
MVI  B/,RLC,,DAD  B,LDAX B,DCX  B,INR  C,DCR  C,2MV
I  C/,RRC
 1120 DATA ,3LXI  D/,STAX D,INX  D,INR  D,DCR  D,2M
VI  D/,RAL,,DAD  D,LDAX D,DCX  D,INR  E,DCR  E,2MVI
  E/,RAR
 1130 DATA RIM,3LXI  H/,"3SHLD ",INX  H,INR  H,DCR
 H,2MVI  H/,DAA,,DAD  H,"3LHLD ",DCX  H,INR  L,DCR
 L,2MVI  L/,CMA
```

```
 1140 DATA SIM,3LXI  SP/,"3STA  ",INX  SP,INR  M,DC
R  M,2MVI  M/,STC,,DAD  SP,"3LDA  ",DCX  SP,INR  A,
DCR  A,2MVI  A/,CMC
 1150 '
 1160 DATA ADD,ADC,SUB,SBB,ANA,XRA,ORA,CMP
 1170 '
 1180 DATA RNZ,POP  B,"3JNZ  ","3JMP  ","3CNZ  ",PU
SH B,"2ADI  ",RST  0,RZ,RET,"3JZ  ",,"3CZ  ","3CA
LL ","2ACI  ",RST 1
 1190 DATA RNC,POP  D,"3JNC  ","2OUT  ","3CNC  ",PU
SH D,"2SUI  ",RST  2,RC,,"3JC  ","2IN  ","3CC  "
,,"2SBI  ",RST  3
 1200 DATA RPO,POP  H,"3JPO  ",XTHL,"3CPO  ",PUSH H
,"2ANI  ",RST  4,RPE,PCHL,"3JPE  ",XCHG,"3CPE  ",,"
2XRI  ",RST  5
 1210 DATA RP,POP  PSW,"3JP  ",DI,"3CP  ",PUSH PS
W,"2ORI  ",RST  6,RM,SPHL,"3JM  ",EI,"3CM  ",,"2C
PI  ",RST  7
 1220 '
 1230 DATA B,C,D,E,H,L,(HL),A
```

DEBUG

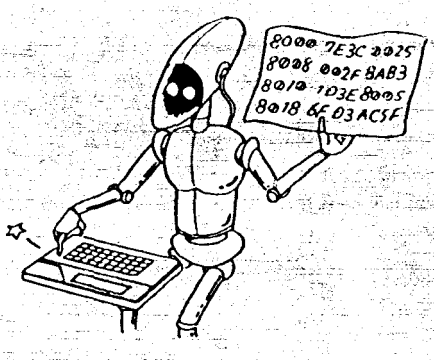
When used in combination with the hexadecimal dump program (see the following chapter), this program helps in understanding machine language.

HEXADECIMAL DUMP

A program for a memory dump in machine language.

Program Outline

Since no machine language monitor is available with the FP–200, the PEEK command is used to check the contents of memory. Still, a memory dump capability is required when reading the contents of memory or preparing a program in machine language.
This program, written in BASIC, provides a memory dump capability.



Data Input

When the program is started, it asks whether or not a printer is used. If you use a printer, press the Y key. If not, press the N key.
Next, input the starting and ending addresses of the memory area to be dumped, in hexadecimal. If the ending address is smaller than the starting address, the program prompts for re-input of the appropriate data.
When a printer is used, the program continuously outputs 16 bytes of memory contents (hexadecimal) per line, together with the associated character codes and check-sum, on the printer immediately after the addresses are input.
When a printer is not used, the program displays 16 bytes of memory contents (hexadecimal) on four lines, together with the associated character codes and check-sum below the last line. Pressing any key on the keyboard causes the next 16 bytes of memory contents to be displayed. Repeat this operation until all the memory contents up to the specified ending address have been displayed.
At the end of the dump operation, the program again prompts for input.

Equation for calculating check sum:

when,

0100 CA 06 01 ..... 76 12

address — 16 bytes of memory contents

0100+CA+06+01+.....+76+12=609 (hexadecimal)

lower 2 digits of 609 → check sum is 09.

HEXADECIMAL DUMP

Example of output when printer is used

```
addr    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ascii              sum
-----+-----------------------+-----------------------+------------------+---
0000  31 10 83 3E CA C3 81 00  1A FE 20 C0 13 C3 08 00  1.. ... . .....    E6
0010  7C BA C0 7D BB C9 00 C3  7E 23 66 6F C9 00 00 00  |..}....~#fo....    09
0020  CF FE 03 D8 C3 5C 10 E1  C3 6C 24 C9 C3 0C 12 C3  .....¥...l$.....    8E
0030  C3 7B 00 9A C3 E8 85 21  C3 91 37 00 E5 D5 C5 F5  .{.....!..7.....    58
0040  CD B3 01 D2 70 00 CD 13  83 CD 6E 03 14 C2 5C 00  ....p.....n...¥.    D6
0050  7E A7 CA 70 00 CD 7D 0F  23 C3 50 00 14 CC 61 04  ~..p..}.#.P...a.    83
0060  14 CC 62 04 14 CC 5B 04  67 7A FE 10 7C D4 7D 0F  ..b...[.gz..|.}.    B0
0070  F1 C1 D1 E1 FB C9 F7 C8  C3 5C 10 CF FE 2C C0 13  .........¥.....    52

0080  C9 30 3E 04 D3 43 3E 40  D3 08 3E 02 D3 11 3E 4A  .0...C>@..>...J    D6
0090  30 AF D3 1E 3E CC D3 1F  3E 99 D3 1D 3E 11 D3 1A  0...>...>...>...    5F
00A0  3E CA 30 CD 43 11 3E 06  32 12 83 AF 32 42 86 32  >.0.C.>.2...2B.2    DF
00B0  EB 87 32 2A 83 CD A3 12  CD 17 13 CD C5 65 CD 22  ..2*.........e."    60
00C0  5F CD CB 60 CD D4 13 21  13 83 06 15 3E C9 CD 39  _..`....!....>..9    AA
00D0  75 3A 11 83 3C CA E0 00  3A 00 E0 FE C3 CC 00 E0  u:..<...:.......    80
00E0  F3 CD 43 11 CD 29 0D CD  F3 0E AF 32 57 83 32 A9  ..C..).....2W.2.    5B
00F0  84 32 AA 84 CD 03 02 E6  01 32 A6 84 F5 21 42 12  .2.......2...!B.    53
```

Example of output when printer is not used

```
0000  31 10 83 3E
      CA C3 81 00
      1A FE 20 C0
      13 C3 08 00
    1..>.....  .....
           sum E6
```

Press any key on the keyboard.

```
0010  7C BA C0 7D
      BB C9 00 C3
      7E 23 66 6F
      C9 00 00 00
    |..}....~#fo.....
           sum 09
```

LIST

1455 Bytes

```
  100 ' initiarize
  110 CLS:PRINT "* Hex-Dump FP-200 * "
  120 INPUT "Printer (Y/N):",Z$:IF Z$="Y" OR Z$="y"
 THEN PSW=1 ELSE PSW=0
  200 '
  210 PRINT :INPUT "Top addr(HEX):",AD$:GOSUB 1000:
ADTOP=AD:IF ER=1 THEN 300
  220 INPUT "End addr(HEX):",AD$:GOSUB 1000:ADEND=A
D:IF ER=1 THEN 300
  230 IF ADTOP>=ADEND THEN 200 ELSE C=0
  240 FOR J=ADTOP TO ADEND STEP 16:H$="":C$="":SUM=
J:X=J:L=4:GOSUB 1200:H$=X$+"  "
  250 FOR I=0 TO 15:M=PEEK (J+I):GOSUB 1100:H$=H$+M
$+" ":C$=C$+A$:SUM=SUM+M:NEXT I
  260 X=SUM:L=2:GOSUB 1200:S$=X$:IF C=0 THEN GOSUB
400
  270 IF PSW=1 THEN LPRINT H$;" ";C$;"  ";S$:GOSUB
500:GOTO 290
  280 PRINT LEFT$(H$,17);TAB(6);MID$(H$,19,12);TAB(
6);MID$(H$,32,11);TAB(6);MID$(H$,44):PRINT TAB(4);C
$
  290 NEXT J:GOTO 200
  300 '
  310 IF PSW=1 THEN LPRINT CHR$(12);
  320 END
  400 '
  410 IF PSW=1 THEN 440
  420 IF J=ADTOP THEN CLS:RETURN
  430 IF INKEY$="" THEN 430 ELSE CLS:RETURN
  440 FOR I=1 TO 3:LPRINT:NEXT I
  450 LPRINT "addr   0  1  2  3  4  5  6  7   8  9
 A  B  C  D  E  F  ascii              sum"
  460 LPRINT "-----+-----------------------------------
-------------------+--------------------+----"
  470 RETURN
  500 '
  510 C=C+1:IF C=8 THEN LPRINT
  520 IF C=16 THEN C=0
  530 RETURN
 1000 ' HEX$(AD$) to DEC(AD)
 1010 AD=0:Y=1:ER=0
```

```
 1020 FOR I=1 TO LEN(AD$):W=ASC(RIGHT$(AD$,I))-48:I
F W<0 OR W>=10 AND W<=16 OR W>22 THEN ER=1:GOTO 106
0
 1030 IF W>16 THEN W=W-7
 1040 AD=AD+W*Y:Y=Y*16
 1050 NEXT I
 1060 RETURN
 1100 ' memory dec(M) to hex$(M$)
 1110 X=M:L=2:GOSUB 1200:M$=X$:IF I=7 THEN M$=M$+"
"
 1120 A$=CHR$(M):IF M<32 OR M>126 THEN A$="."
 1130 RETURN
 1200 'DEC(J) to HEX$(X$)
 1210 X$=""
 1220 Y=INT(X/16):W=X-Y*16:X=Y:IF W>9 THEN W=W+7
 1230 X$=CHR$(W+48)+X$:IF X<>0 THEN 1220
 1240 X$=RIGHT$("000"+X$,L)
 1250 RETURN
```

DEBUG

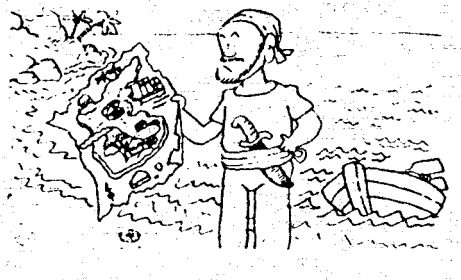
The routine to read the contents of memory is contained in line 250. Although this program displays output data in hexadecimal notation, actual operations are performed on decimal numbers. Hence a hexadecimal-decimal conversion program is required. The three subroutines in lines 1000 to 1250 are provided for this purpose.

Program

TREASURE HUNTING

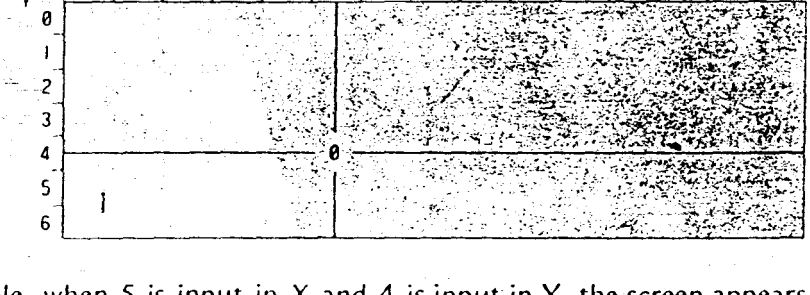
Program Outline

This is a treasure-hunting game. The number of treasures can be specified in the range from 1 to 9. The point of this game is to get the specified number of treasures with the minimum number of key strokes. As you manipulate keys, hints are given by numbers to help you find the treasures.



Directions

The specified number of treaures are hidden somewhere in 15 x 7 matrices.
When the program is started, it asks for the number of treasures to be hidden. Input any number from 1 to 9. The greater the number, the more difficult the game becomes. At first, try it with 3 to 5 treasures.
The program then asks for the values of X (horizontal axis) and Y (vertical axis). Input the appropriate values. The maximum value of X is 6, and the maximum value of Y is 14. Then, the program draws lines to indicate the data at the intersecting point of X and Y on the screen.



For example, when 5 is input in X and 4 is input in Y, the screen appears as shown in the diagram, above. At the intersecting point, the following is displayed:

0 → No treasure can be found on lines X=5 or Y=4.
1 → A treasure can be found on either of the two lines.
2 → Two treasures can be found on either or both of the two lines.
3 → A total of three treasures can be found on either or both of the two lines.
4—15 → The Corresponding number of treasures can be found on either or both of the two lines.
$ → A treasure is located at the intersecting point.

When you find all the hidden treasures, the program outputs the scores. The scores indicate the number of key-strokes required to find all the treasures. Thus, the lower the scores, the better.

LIST *886 Bytes*

```
 10 CLS:CLEAR:DIM A(14,7):DIM B$(14,7)
 20 INPUT "1-15";P:CLS
 30 RX=INT(RND(-1)*15):RY=INT(RND(-1)*8)
 40 IF A(RX,RY)<>1 THEN A(RX,RY)=1:C=C+1
 50 IF C<P THEN 30
 60 QUAD (8,0)-(128,56)
 70 LOCATE 17,1:INPUT"X",X$:X=VAL(X$)
 80 LOCATE 17,2:INPUT"Y",Y$:Y=VAL(Y$)
 85 IFX$="" OR Y$="" THEN 75
 90 IF X<0 OR X>14 OR Y<0 OR Y>6 THEN 70
100 IF A(X,Y)=2 THEN70
110 X=X+1
120 XP=(X+1)*8-4:YP=(Y+1)*8-4
130 IF A(X-1,Y)=1 THENAT=AT+1:A(X-1,Y)=2:LOCATE X
,Y:PRINT "$":B$(X-1,Y)=" $":GOTO 240
140 DRAW (XP,0):DRAW -(XP,56)
150 DRAW (8,YP):DRAW -(128,YP)
160 FOR I=0 TO 6
170 IF A(X-1,I)>=1 THEN CO=CO+1
180 NEXT I:FOR I=0 TO 14
190 IF A(I,Y)>=1 THEN CO=CO+1
200 NEXT I
210 IF X=0 THEN LOCATE 0,Y:PRINT CO:B$(0,Y)=STR$(
CO):GOTO 240
220 LOCATE (X-1),Y:PRINT CO:B$(X-1,Y)=STR$(CO)
230 FOR I=1 TO 300:NEXT I
240 CLS
250 T1=T1+1:CO=0:GOSUB 290
260 IF AT>=P THEN 280
270 GOTO 60
280 CLS:LOCATE 5,3:PRINT "SCORE ";T1:END
290 FOR I=6 TO 0 STEP -1:FOR J=14TO 0 STEP -1:LOC
ATE 0,I
300 PRINT TAB(J);B$(J,I);
310 NEXT J:NEXT I
320 RETURN
```

DEBUG

The display of a $ indicating that a treasure is found is done in line 130. Lines 160 to 230 search for treasures and indicate the number of treasures that can be found on particular lines. If an FC error occurs, execute 'INIT (0,0), 1,1' and press the ↵ key.

POKER GAME

Program Outline

This is a poker game played against the computer. Can you beat the 'poker-faced' computer?

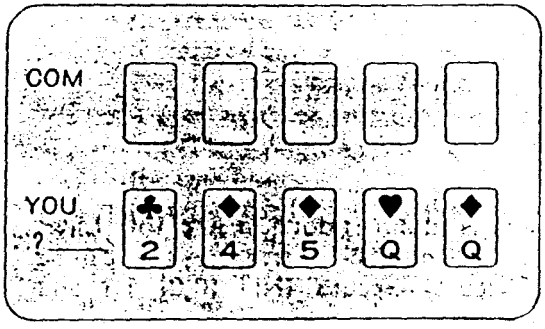
Directions

When the program is started, it displays the amount of your playing money, then asks for the amount you want to bet. Your stake is 10. You can bet only up to your stake. When you input the bet, the program displays "JUST A MOMENT," then lays out cards. Next, the COM (computer) cards (face down) and your cards (face up) are displayed. You can draw cards only once! If you wish to draw a card (or cards), specify position of the card, counted from the left. You may draw any number of cards.
After you draw cards, the cards are displayed again and the winner is determined. The game is over if you lose your money.

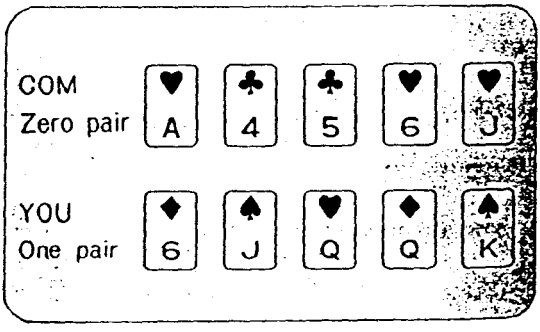
Rate table

Hand	Multiplier	Hand	Multiplier
One pair		Flush	10 times
Two pairs	2 times	Four of a kind	30 times
Three of a kind	5 times	Royal straight	40 times
Full house	8 times	Straight flush	150 times
Straight	10 times	Royal flush	600 times

Drawing cards



Who wins? Hands are displayed in order of strength from the left



LIST

3299 Bytes

```
 10 CLS:CLEAR
 20 DIM L(5),CA$(20),DA$(20),CA(20),DA(20),CD$(20

 30 C$="A23456789TJQK":D$="♦♣♠♥"
 40 A=10
 50 LOCATE 5,2:PRINT "Stake";A
 60 LOCATE 5,4:INPUT "Bet";A1:IF A-A1<0 THEN 60
 70 CLS:LOCATE 4,3:PRINT "JUST A MOMENT":YA=0:YB=
0:YC=0:YD=0:YE=0:Z=0
 80 GOSUB 940:CLS:GOSUB 700
 90 LOCATE 0,2:PRINT "COM":LOCATE 0,5:PRINT "YOU"
100 QUAD(34,4)-(54,27)
110 QUAD(58,4)-(78,27)
120 QUAD(82,4)-(102,27)
130 QUAD(106,4)-(126,27)
140 QUAD(130,4)-(150,27)
150 QUAD(34,29)-(54,52)
160 QUAD(58,29)-(78,52)
170 QUAD(82,29)-(102,52)
180 QUAD(106,29)-(126,52)
190 QUAD(130,29)-(150,52)
200 LOCATE 5,4:PRINT DA$(6):LOCATE 8,4:PRINT DA$(
7):LOCATE 11,4:PRINT DA$(8):LOCATE 14,4:PRINT DA$(9
):LOCATE 17,4:PRINT DA$(10)
210 LOCATE 5,5:PRINT CA$(6):LOCATE 8,5:PRINT CA$(
7):LOCATE 11,5:PRINT CA$(8):LOCATE 14,5:PRINT CA$(9
):LOCATE 17,5:PRINT CA$(10)
220 IF M=1 THEN 280
230 LOCATE 0,6:INPUT HN:E=10:I=11
240 H=INT((FRAC(HN/E))*10):IF H>5 THEN 230
250 IF H=0 THEN 270
260 H=H+5:CA$(H)=CA$(I):DA$(H)=DA$(I):CA(H)=CA(I)
:I=I+1:E=E*10:GOTO 240
270 GOSUB 700:M=1:CLS:GOTO 90
280 FOR I=7 TO 10
290 IF DA$(6)<>DA$(I) THEN 310
300 NEXT I:YA=6
310 FOR I=6 TO 9
320 IF CA(I)+1<>CA(I+1) THEN 350
330 NEXT I:IF YA=6 THEN YA=9:GOTO 560
```

```
 340 YA=5:GOTO 560
 350 IF CA$(6)<>"A"  THEN 420
 360 IF CA$(7)<>"10" THEN 420
 370 IF CA$(8)<>"J"  THEN 420
 380 IF CA$(9)<>"Q"  THEN 420
 390 IF CA$(10)<>"K" THEN 420
 400 IF YA=6 THEN YA=10:GOTO 560
 410 YA=8:GOTO 560
 420 IF YA=6 THEN 560
 430 FOR I=6 TO 9:IF CA$(I)=CA$(I+1) THEN YB=YB+1
 440 NEXT I:IF YB<>3 THEN 480
 450 IF CA$(6)<>CA$(7)  THEN YA=7:GOTO 560
 460 IF CA$(9)<>CA$(10) THEN YA=7:GOTO 560
 470 YA=4:GOTO 560
 480 IF YB=0 THEN YA=0:GOTO 560
 490 IF YB=1 THEN YA=1:GOTO 560
 500 FOR I=6 TO 9
 510 IF CA$(I)=CA$(I+1) THEN YC=YC+1:GOTO 530
 520 YC=0
 530 IF YC=2 THEN YA=3:GOTO 560
 540 NEXT I
 550 YA=2
 560 IF Z=1 THEN LOCATE 1,3 ELSE LOCATE 1,6
 570 IF YA=0  THEN PRINT "Zero pair"
 580 IF YA=1  THEN PRINT "One pair"
 590 IF YA=2  THEN PRINT "Two pairs"
 600 IF YA=3  THEN PRINT "Three of a kind"
 610 IF YA=4  THEN PRINT "Full house"
 620 IF YA=5  THEN PRINT "Straight"
 630 IF YA=6  THEN PRINT "Frash"
 640 IF YA=7  THEN PRINT "Four of a kind"
 650 IF YA=8  THEN PRINT "Royal straight"
 660 IF YA=9  THEN PRINT "Straight frash"
 670 IF YA=10 THEN PRINT "Royal straight frash"
 680 GOTO 740
 690 '
 700 FOR I=9 TO 6 STEP -1:FOR J=6 TO I
 710 IF CA(J)>CA(J+1) THEN O=CA(J):CA(J)=CA(J+1):C
A(J+1)=O:P$=CA$(J):CA$(J)=CA$(J+1):CA$(J+1)=P$:P$=D
A$(J):DA$(J)=DA$(J+1):DA$(J+1)=P$
 720 NEXT J:NEXT I:RETURN
 730 '
 740 IF Z<>0 THEN FOR I=0 TO 2000:NEXT I:GOTO 790
 750 YT=YA:YA=0:YB=0:YC=0:FOR I=1 TO 5
```

```
  760 CA$(I+5)=CA$(I):CA(I+5)=CA(I):DA$(I+5)=DA$(I)
:NEXT I:Z=1:GOSUB 700
  770 LOCATE 5,1:PRINT DA$(6):LOCATE 8,1:PRINT DA$(
7):LOCATE 11,1:PRINT DA$(8):LOCATE 14,1:PRINT DA$(9
):LOCATE 17,1:PRINT DA$(10):M=0
  780 LOCATE 5,2:PRINT CA$(6):LOCATE 8,2:PRINT CA$(
7):LOCATE 11,2:PRINT CA$(8):LOCATE 14,2:PRINT CA$(9
):LOCATE 17,2:PRINT CA$(10):FOR I=1 TO 600:NEXT I:G
OTO 280
  790 IF YT<=YA THEN CLS:LOCATE 3,3:PRINT "COMPUTER
 WIN !!":FOR I=1 TO 800:NEXT I:A=A-A1:GOTO 910
  800 CLS:LOCATE 4,3:PRINT "YOU WIN !!":FOR I=1 TO
800:NEXT I
  810 IF YT=1  THEN A=A+A1
  820 IF YT=2  THEN A=A+A1*2
  830 IF YT=3  THEN A=A+A1*5
  840 IF YT=4  THEN A=A+A1*8
  850 IF YT=5  THEN A=A+A1*10
  860 IF YT=6  THEN A=A+A1*15
  870 IF YT=7  THEN A=A+A1*30
  880 IF YT=8  THEN A=A+A1*40
  890 IF YT=9  THEN A=A+A1*150
  900 IF YT=10 THEN A=A+A1*600
  910 IF A=0 THEN LOCATE 1,3:PRINT "YOU HAVE NO MON
EY":LOCATE 3,5:PRINT "GAME OVER ":END
  920 CLS:FOR I=0 TO 20:CA(I)=0:DA(I)=0:CD$(I)="":N
EXT I:GOTO 50
  930
  940 FOR I=1 TO 20
  950 CA(I)=INT(RND(-1)*13+1)
  960 CA$(I)=MID$(C$,CA(I),1):IF CA$(I)="T" THEN CA
$(I)="10"
  970 DA(I)=INT(RND(-1)*4+1)
  980 DA$(I)=MID$(D$,DA(I),1):CD$=DA$(I)+CA$(I)
  990 FOR J=I TO 1 STEP -1:IF CD$=CD$(J) THEN 950
 1000 NEXT J:CD$(I)=CD$:NEXT I:RETURN
```

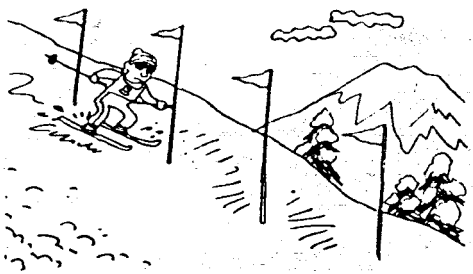
DEBUG

CA and CA$ are variables for dealing cards by random numbers; DA and DA$ are variables for ♣ – ♠ marks; H and HN are for changing cards; and YA, YB, and YC are variables for determining the strengths of hands.

SKI SLALOM

Program Outline

Skiing is one of the most popular winter sports. It is really refreshing to thread your way downhill. This game offers some of the fun of skiing even in your living room.



Directions

When the program is started, it displays "SKI GAME." It also displays a message telling you to press the K key to move to the left and the L key to move to the right. To start the game, press any key on the keyboard. The skier is indicated by a clover leaf symbol (♣). Using the two keys (K and L), thread your way between woods and holes. You can try three times. Each time you reach the goal, the scores are displayed as POINTS. When you fail three times, the total score is displayed. The program then asks whether or not you wish to try again. If so, press the Y key.

LIST

663 Bytes

```
    10 CLS:CLEAR:M=3
    15 CLS:LOCATE 4,2:PRINT"K ▲ ♣ ▲ L":PRINT:PRINT
"       SKI GAME"
    16 IF INKEY$="" THEN 16
    20 X=10:Y=1:P=1:CLS
    30 J=INT(RND(-1)*20+1):K=INT(RND(-1)*20+1):L=INT
(RND(-1)*20+1)
    33 Y$=INKEY$
    35 IF Y$="K" THEN P=-1
    36 IF Y$="L" THEN P=1
    45 LOCATE (J-1),7:PRINT "■";:LOCATE (K-1),7:PRIN
T"■";:LOCATE (L-1),7:PRINT"■"
    46 IF X>19 THEN X=19
    47 IF X<2 THEN X=2
    48 LOCATE 0,Y:PRINT TAB(X-1);"♣";:PO=PO+10
    50 X1=X*8-4:Y1=(Y+1)*8+4:IF P=1 THEN X2=X*8+4:Y2
=(Y+1)*8+4
    60 IF P=-1 THEN X2=X*8-12:Y2=(Y+1)*8+4
    70 IF ((POINT(X1,Y1)=-1)OR(POINT(X2,Y2)=-1)) THE
N 300
    80 X=X+P:GOTO 30
   300 CLS:LOCATE 4,4:PRINT "POINT:";PO:M=M-1:GOSUB
1000:IF M>0 THEN CLS:GOTO 20
   310 CLS:LOCATE 2,3:INPUT"PLAY AGAIN(Y)";A$:IF A$=
"Y" THEN 10
   320 END
  1000 FOR I=1 TO 300:NEXT I:RETURN
  1010 END
```

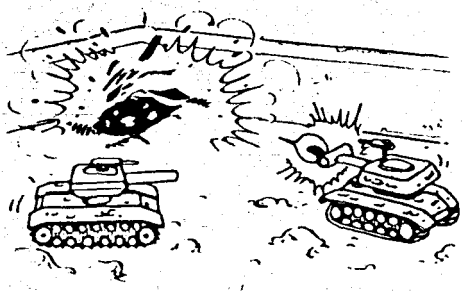
DEBUG

The game execution routine is contained in lines 30 to 80. Line 45 displays obstacles. Line 48 displays the skier as a clover leaf symbol.

FIGHTING ARMOR

Program Outline

You are the commander of a tank located on a square combat area. Move about freely within the combat area to destroy the enemy. Note that the energy pellet of your tank lasts for only one and a half minutes. How many opposing tanks can you destroy within the specified time?



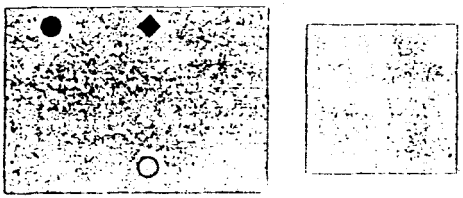
Directions

Run the program, and the combat area appears on the left side of the screen and the target sight appears on the right side of the screen. The white circle indicates your tank, and the solid circle indicates the enemy. The keys I, J, L, and M are used to move up, move left, move right, and move down, respectively. Move the white circle so that it is on the same line as the solid circle (see ❷). Then, a + (sight) and a # (enemy position) appear in the target sight. Move the sight so that it is on the #. When you succeed in doing so, you destroy the enemy. To adjust the sight, use the keys i (shifted I), j (shifted J), l (shifted L), and m (shifted M); i for moving up, j for moving left, l for moving right, and m for moving down.

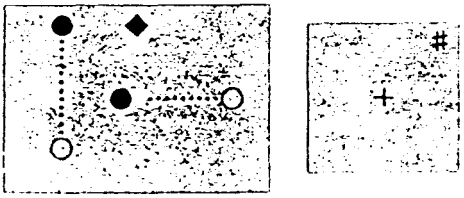
The enemy also attacks you from the ◆ in the combat area. When the ◆ turns into a ■, the ■ will soon move to the current position of your tank. You should immediately move away from the current position, or you will be destroyed by the enemy. You are also destroyed if your white circle hits the solid circle or the stationary ◆ or ■. The game is over when you are destroyed three times or when the specified time (1.5 minutes) passes. Then the scores are displayed.

Display

❶



❷



LIST *2524 Bytes*

```
   10 CLEAR:TIME$="00:00:00":DIM A$(10,6)
   20 CLS:TX=5:TY=6:SX=15:SY=3:TX2=13:TY2=1:W=0:XX=
5:YY=0:XXX=5:YYY=0:CT=0
   30 TX2=INT(RND(-1)*5)+13:TY2=INT(RND(-1)*5)+1
   40 LOCATE 5,0:PRINT"◆"
   50 R=INT(RND(-1)*4)
   60 IF R=0 THEN TX1=0:TY1=0
   70 IF R=1 THEN TX1=10:TY1=0
   80 IF R=2 THEN TX1=0:TY1=3
   90 IF R=3 THEN TX1=10:TY1=3
  100 LOCATE TX1,TY1:PRINT"●"
  110 LOCATE TX,TY:PRINT"○"
  120 QUAD (0,0)-(88,56):QUAD (104,8)-(144,48)
  130 K$=INKEY$:LOCATE TX,TY:PRINT" ":LOCATE TX1,TY
1:PRINT" "
  140 IF K$="I" THEN TY=TY-1
  150 IF K$="M" THEN TY=TY+1
  160 IF K$="J" THEN TX=TX-1
  170 IF K$="L" THEN TX=TX+1
  180 IF TY<0 THEN TY=0
  190 IF TY>6 THEN TY=6
  200 IF TX<0 THEN TX=0
  210 IF TX>10 THEN TX=10
  220 LOCATE TX,TY:PRINT"○"
  230 R=INT(RND(-1)*5):IF R<4 THEN 320
  240 IF TX1=TX THEN IF TY1>TY THEN TY1=TY1-1:GOTO
280:ELSE TY1=TY1+1:GOTO 280
  250 IF TY1=TY THEN IF TX1>TX THEN TX1=TX1-1:GOTO
280:ELSE TX1=TX1+1:GOTO 280
  260 IF TX1>TX THEN TX1=TX1-1:GOTO 280 ELSE TX1=TX
1+1:GOTO 280
  270 IF TY1>TY THEN TY1=TY1-1 ELSE TY1=TY1+1
  280 IF TY<0 THEN TY=0
  290 IF TY>7 THEN TY=7
  300 IF TX<0 THEN TX=0
  310 IF TX>10 THEN TX=10
  320 LOCATE TX1,TY1:PRINT"●"
  330 GOSUB 490
  340 GOSUB 460
```

```
  350 R=INT(RND(-1)*9):IF R=3 AND W=0 THEN GOSUB 45
0
  360 IF W=1 THEN CT=CT+1
  370 IF CT>=6 THEN LOCATE XXX,YYY:PRINT" ":LOCATE
XX,YY:PRINT"■":XXX=XX:YYY=YY:W=0
  380 IF CT>=6 AND XX=TX AND YY=TY THEN LOCATE XX,Y
Y:PRINT"*":FOR I=1 TO 300:NEXT I:CLS:LOCATE 2,3:PRI
NT"YOU DESTROYED!":FOR I=1 TO 300:NEXT I:AT=AT+1:IF
 AT<=2 THEN 20 ELSE 780
  390 LOCATE XX,YY:PRINT" "
  400 PP=POINT (XX*8+4,YY*8+4)
  410 IF W=0 AND PP=0 THEN LOCATE XX,YY:PRINT"◆"
  420 IF CT>=6 THEN CT=0
  430 IF TIME$>"00:01:30" THEN 780
  440 GOTO 130
  450 W=1:XX=TX:YY=TY:LOCATE XXX,YYY:PRINT"■":RETUR
N
  460 IF TX1=TX THEN GOSUB 510
  470 IF TY1=TY THEN GOSUB 510
  480 RETURN
  490 IF(TX1=TX AND TY1=TY)OR(XXX=TX AND YYY=TY)THE
N AT=AT+1:LOCATE TX,TY:PRINT"*":FOR I=1 TO 300:NEXT
 I:CLS:LOCATE 2,3:PRINT"YOU DESTROYED!":FOR I=1 TO
300:NEXT I:IF AT<=2 THEN GOTO 20 ELSE 780
  500 RETURN
  510 REM
  520 SXS=SX:SYS=SY:TX2S=TX2:TY2S=TY2
  530 QUAD (104,8)-(144,48):LOCATE TX2,TY2:PRINT"#"
:LOCATE SX,SY:PRINT"+"
  540 E$=INKEY$
  550 IF SX=TX2 AND SY=TY2 THEN BK=BK+1:LOCATE TX1,
TY1:PRINT"*":FOR I=1 TO 300:NEXT I:CLS:LOCATE 2,3:P
RINT"ENEMY DESTROYED!":FOR I=1 TO 300:NEXT I:GOTO 2
0
  560 IF E$="i" THEN SY=SY-1:GOTO 720
  570 IF E$="m" THEN SY=SY+1:GOTO 720
  580 IF E$="j" THEN SX=SX-1:GOTO 720
  590 IF E$="l" THEN SX=SX+1:GOTO 720
  600 Q=INT(RND(-1)*3):IF Q=0 THEN 710
  610 R=INT(RND(-1)*4)
  620 IF R=0 THEN TX2=TX2-1
  630 IF R=1 THEN TX2=TX2+1
  640 IF R=2 THEN TY2=TY2-1
  650 IF R=3 THEN TY2=TY2+1
  660 IF TX2<13 THEN TX2=13
```

```
670 IF TX2>17 THEN TX2=17
680 IF TY2<1 THEN TY2=1
690 IF TY2>5 THEN TY2=5
700 GOSUB 770
710 LOCATE SXS,SYS:PRINT" ":RETURN
720 IF SX<13 THEN SX=13
730 IF SX>17 THEN SX=17
740 IF SY<1 THEN SY=1
750 IF SY>5 THEN SY=5
760 GOTO 600
770 LOCATE SXS,SYS:PRINT" ":LOCATE TX2S,TY2S:PRIN
T" ":RETURN
780 CLS:PRINT"*** GAME OVER ***":LOCATE 3,3:PRINT
" SCORE=";BK:END
```

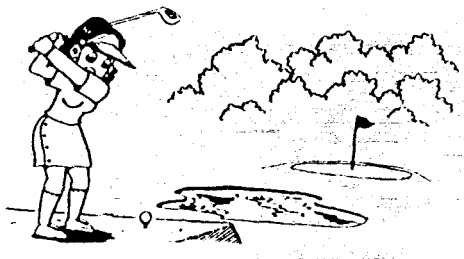
DEBUG

The INKEY$ function of the FP–200 obtains one character of input from the buffer. Hence, only one character is read even if the key is kept depressed. Pressing the key several times causes the same number of moves as the keystrokes to be made subsequently. In order to prevent this, it is advisable to manipulate the key while observing the screen display.

GOLF GAME

Program Outline

If you are a golf enthusiast, you must have had the disappointing experience of your golf game being suspended due to bad weather. It's no use crying over spilt milk. Let's play a golf game for you. It has nine holes, and you will not find it so easy to get good scores. If you are a novice, you will discover some of the fun of golfing.



Directions

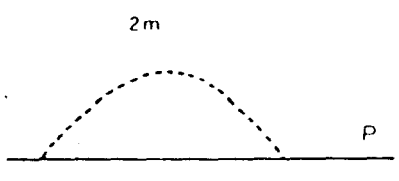
Start the program, and a ♠ (bushes), ● (water hazard), and P (pin) are displayed on the screen. The tee is positioned at the left end of the screen. The prompt "POWER (1—10)" is displayed at the bottom of the screen. Input a number from 1 to 9. If a negative number is input, the ball flies in the opposite direction. If the ball goes into the water hazard, a one-shot penalty is added. If you go into the bushes, the appropriate penalty is imposed. A power of 1 moves the ball by approximately one character position (no wind). Wind data is displayed on top of the screen: the arrowhead indicates the wind direction and the number indicates the wind velocity. Both exert a considerable influence on the distance of flight.

When the ball drops near the pin (P), the screen turns green. Characters U, D, R, and L which appear on the screen indicate the condition of the green: U indicates a downward slope toward the top of the screen; D, toward the bottom of the screen; R, to the right; and L, to the left. The figures shown indicate the angle of the slope. The power on the green should be from 0 to 6, and a power of 1 moves the ball by about 10 dots. When the ball gets in the cup (●), the screen displays a birdie, par, bogey, etc., then displays the next hole.

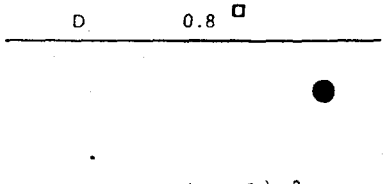
After you go around the nine holes, the final scores are displayed. If you wish to play again, press the Y key. In order to get good scores, you should play again and again. Note that inputting an extremely large number in POWER may cause the ball to fly out of sight.

Display

Fair way



Putting green



LIST

2632 Bytes

```
   10 CLEAR: DIM CD(9),PD(9)
   20 CLS:LOCATE 3,3:PRINT"THE GOLF GAME":GOSUB 650
   30 FOR J=1 TO 9:READ CD(J):READ PA(J):READ PD(J)
:NEXT J
   40 FOR HO=1 TO 9:CLS:LOCATE 6,3:PRINT HO;" Hole"
:LOCATE 10,5:PRINT"Par";PA(HO)
   50 PT=0:X=0:Y=40
   60 GOSUB 650:CLS:DRAW(0,40):DRAW-(159,40)
   70 BU=INT(RND(-1)*14+5):IF BU=PD(HO) THEN 70
   80 BA=INT(RND(-1)*14+5):IF BA=BU THEN 80
   90 IF BA=PD(HO) THEN 80
  100 LOCATE (BU-1),4:PRINT" ":LOCATE (BA-1),4:PRIN
T" ":LOCATE (PD(HO)-1),4:PRINT"P"
  110 IF RND(-1)<0.5 THEN WI=INT(RND(-1)*10):LOCATE
 0,0:PRINT"◀";WI/2;"m":DR=0:GOTO 130
  120 WI=INT(RND(-1)*10):LOCATE 0,0:PRINT"▶";WI/2;
"m":DR=1
  130 LOCATE 1,6:INPUT"Power(1~10)?",PW:PT=PT+1:IF
PW>10 THEN 130
  140 IF PW<0 THEN ST=-1:GOTO 160
  150 ST=1
  160 PW=INT(PW*8):IF DR=0 THEN PW=INT(PW+WI*2+RND(
-1)*5):GOTO 180
  170 PW=INT(PW-WI*2-RND(-1)*5)
  180 FOR R=0 TO 180 STEP (180/ABS(PW)):X=INT(X+ST)
:Y1=Y-SIN(R)*24:IF ((X<0) OR (X>159)) THEN LOCATE 1
,6:PRINT"   OB  !!      ":PT=PT+2:GOSUB 650:GOTO 130
  190 DRAW(X,Y1):NEXT R
  200 IF (X<(BA*8) AND X>(BA*8-7)) THEN PT=PT+1:GOT
O 130
  210 IF (X<(BU*8) AND X>(BU*8-7)) THEN LOCATE 1,6:
P=INT(RND(-1)*3):PRINT P;"Plus       ":GOSUB 650:PT=
PT+P:GOTO 130
  220 IF X<((PD(HO)-1)*8) THEN 130
  230 IF X>(PD(HO)*8) THEN 130
  240 CLS:LOCATE 3,3:PRINT"Green on !":GOSUB 650:CL
S
  250 GR=ROUND(RND(-1)*3,-2):KI=INT(RND(-1)*4)
  260 IF KI=0 THEN A$="R":GOTO 300
  270 IF KI=1 THEN A$="U":GOTO 300
  280 IF KI=2 THEN A$="L":GOTO 300
```

```
  290 IF KI=3 THEN A$="D"
  300 LOCATE 1,0:PRINT A$;GR;" "
  310 DRAW(0,8):DRAW-(159,8)
  320 XG=INT(RND(-1)*159):YG=INT(RND(-1)*38+9)
  330 LOCATE 9,3:PRINT"●":DRAW(XG,YG)
  340 LOCATE 1,6:PRINT"                    ":LOCATE 1
,6:INPUT"Power(0~6)?",PW:PT=PT+1:IF ((PW>6) OR (PW<
0)) THEN 340
  350 LOCATE 1,6:PRINT"                    ":LOCATE 1,6:INP
UT"Angle(0~360)?",KA:IF KA>360 THEN 350
  360 FOR I=1 TO INT(PW*5):YG=YG-SIN(KA):XG=XG+COS(
KA)
  370 IF YG<8 THEN YG=9
  380 IF YG>48 THEN YG=47
  390 IF XG<0 THEN XG=1
  400 IF XG>159 THEN XG=159
  410 DRAW(XG,YG):NEXT I
  420 FOR I=1 TO INT(PW*5):YG=YG-SIN(KA):XG=XG+COS(
KA)
  430 IF KI=2 THEN XG=XG-GR/5
  440 IF KI=0 THEN XG=XG+GR/5
  450 IF KI=1 THEN YG=YG-GR/5
  460 IF KI=3 THEN YG=YG+GR/5
  470 IF YG<8 THEN YG=9
  480 IF YG>48 THEN YG=47
  490 IF XG<0 THEN XG=1
  500 IF XG>159 THEN XG=159
  510 IF ((XG>73) AND (XG<80) AND (YG<32)^AND (YG>2
5)) THEN 530
  520 DRAW(XG,YG):NEXT I:GOTO 340
  530 CLS:LOCATE 6,3:PRINT"Cup in !":GOSUB 650
  540 CLS:IF (PT-PA(HO))=-2 THEN LOCATE 5,3:PRINT"E
agle!!":GOSUB 650:GOTO 600
  550 IF (PT-PA(HO))=-1 THEN LOCATE 5,3:PRINT"Birdy
!":GOSUB 650:GOTO 600
  560 IF (PT-PA(HO))=0 THEN LOCATE 5,3:PRINT"Par":G
OSUB 650:GOTO 600
  570 IF (PT-PA(HO))=1 THEN LOCATE 5,3:PRINT"Boggy"
:GOSUB 650:GOTO 600
  580 IF (PT-PA(HO))=2 THEN LOCATE 5,3:PRINT"Double
 boggy":GOSUB 650:GOTO 600
  590 LOCATE 5,3:PRINT"Score:";PT:GOSUB 650
  600 SC=SC+PT:PT=0:NEXT HO
  610 CLS:LOCATE 1,1:PRINT"CASIO country club":LOCA
TE 5,3:PRINT"Par 32":LOCATE 5,5:PRINT"You";SC
```

```
  620 LOCATE 5,6:INPUT"Play again(Y)",A$
  630 IF A$="Y" THEN RESTORE 660:GOTO 10
  640 END
  650 FOR I=0 TO 300:NEXT I:RETURN
  660 DATA 360,4,12,420,4,14,240,3,8,460,4,15,560,5
,19,270,3,9,390,4,13,520,5,17,330,4,11
```

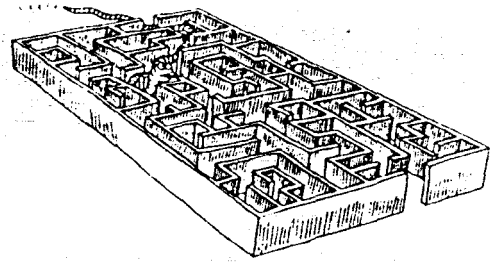
DEBUG

Lines 110 and 120 set the wind conditions. By changing the number that multiplies RND (−1) to an appropriate number, it is possible to adjust the wind intensity. Also, by changing the value 2 in WI*2 in lines 160 and 170, it is possible to adjust the effect of winds on the ball. The decision on OB is made in line 180, and the decision as to whether the ball is in the cup is made in line 510.

MAZING GAME

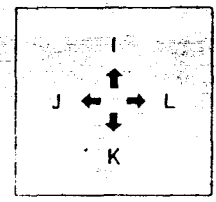
Program Outline

This is an interesting game named the mazing game. You must try to guide the snake through a maze without being blocked.



Directions

Start the program, and it displays the title "MAZING GAME," then asks "LEVEL?". LEVEL (1–5) indicates the number of obstacles placed inside a predetermined area. If level 1 is specified, only one obstacle is placed somewhere in the area; if level 2 is specified, two obstacles are set up, and so on. When LEVEL is specified, an area is set on the screen and the specified number of obstacles are displayed. The 'snake' starts moving in an arbitrary direction. The snake motion should be controlled by the following keys.



Use the I and K keys for vertical movement, and use the J and L keys for horizontal movement. Needless to say, when the snake is moving upward, pressing the I key does not change the snake motion. Note that the snake really moves obliquely toward the top or bottom of the screen. Thus, when the snake is moving toward the top left, the only valid keys are K (move toward the bottom left) and L (move toward the top right). The game is over when you try three times. Your scores and the highest scores of the day are displayed. To play the game again, press the Y key.

LIST

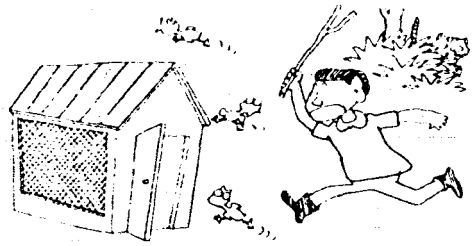
664 Bytes

```
    10 CLS:CLEAR
    25 CLS:LOCATE 2,3:PRINT "MAZE":PRINT:INPUT"    LE
BEL";L:CLS:M=3
    30 IF((L<1)OR(L>5)) THEN 25
    40 X=80:Y=32:DX=1:DY=0:CLS:FOR I=1 TO L
    41 X1=INT(RND(-1)*18+2):Y1=INT(RND(-1)*6+1):IF Y
1=4 THEN 41
    42 LOCATE (X1-1),Y1:PRINT"■":NEXT I
    43 DRAW(0,0):DRAW-(159,0):DRAW-(159,63):DRAW-(0,
63):DRAW-(0,0)
    50 A$=INKEY$:IF A$="L" THEN DX=1
    55 IF A$="I" THEN DY=-1
    60 IF A$="J" THEN DX=-1
    65 IF A$="K" THEN DY=1
    70 X=X+DX:Y=Y+DY
    75 IF POINT(X,Y)=-1 THEN GOTO 200
    80 SC=SC+L:DRAW(X,Y):GOTO 50
   200 CLS:M=M-1:IF M>0 THEN LOCATE 3,3:PRINT"LEFT S
NAKE=";M:GOSUB 1000:GOTO 40
   210 CLS:LOCATE 1,3:PRINT"SCORE:";SC:PRINT" H S";H
S
   213 IF HS<SC THEN HS=SC:PRINT :PRINT"     HI SCORE
!!"
   215 LOCATE 3,6:INPUT" PLAY AGAIN(Y)";Y$:IF Y$="Y"
 THEN 25
   220 END
  1000 FOR I=1 TO 300:NEXT I:RETURN
```

CATCH THE CHICKEN

Program Outline

You try to catch a fat chicken for your dinner, but he makes a desperate attempt to escape from you. Now, you have to devise a good idea to catch him. Whether you will be able to eat a roasted chicken dinner or you will get hurt by a counterattack from the chicken depends solely on your brains.

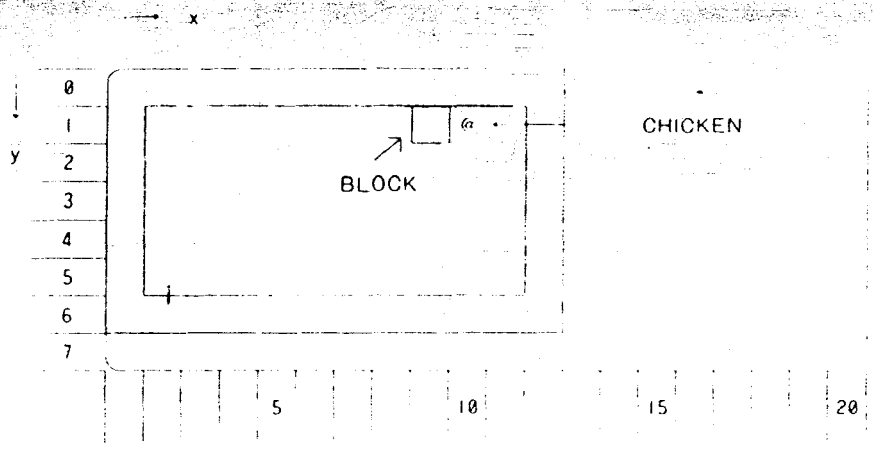


Directions

The chicken is poised in a rectangular fenced-in garden. The chicken, however, runs around the garden when you go in to catch it. You can see that it should be hemmed in so that it can find no retreat.

One thing should be remembered. The chicken has a sharp beak, and if you are pecked with the beak you will get hurt, so you must be careful when attempting to catch it.

The screen appears as shown in the diagram below. The @ inside the rectangle indicates the chicken you are going to catch. You must enclose the chicken with blocks until it can no longer move around. The game is finished when you place a block at the position of the chicken (see diagram, below next page).

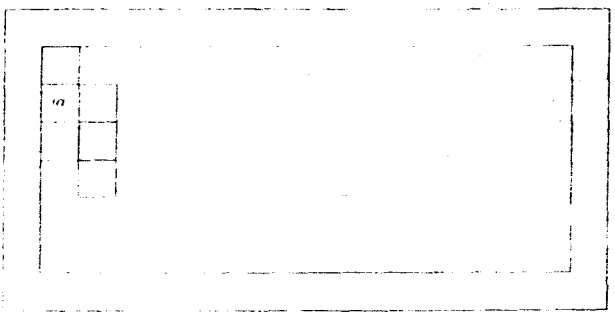


The screen assumes the horizontal direction as the X-axis and the vertical direction as the Y-axis. Thus, in the diagram above the position of the block is represented by coordinates (x=9, y=1). Now you wish to place another block at the position indicated by the dotted lines. Key in 10 when the program asks "X=?–." Then key in 2 when the program asks "Y=?–." This places the block in the desired position.

When you place a block, the chicken moves one position. If the position in which the

block was placed coincides with the position to which the chicken moved, the screen displays "I WIN," indicating that the chicken has won. The game is over. So, you must foresee the possible movements of the chicken in conducting the game. When you successfully catch the chicken, the screen displays "YOU WIN," and the program displays the number of blocks required to catch the chicken. The smaller the number, the better.

When the game is over, the program asks "PLAY AGAIN? (Y)." If you wish to play again, press the Y key.



Now the chicken is completely hemmed in.
You can catch it by keying in x=2, y=2.

LIST *926 Bytes*

```
   10 CLS:CLEAR
   20 LOCATE 0,0:PRINT"■■■
```

```
  120 IF RA=2 THEN IF POINT(X3-8,Y3)<>-1 THEN X2=X2
-1:GOTO 160
  130 IF RA=3 THEN IF POINT(X3,Y3+8)<>-1 THEN Y2=Y2
+1:GOTO 160
  140 C=C+1:IF C<30 THEN 90
  150 CLS:LOCATE 5,4:PRINT"YOU WIN!!":LOCATE 5,6:PR
INT PO;"Times":GOTO 190
  160 PO=PO+1:LOCATE (X1-1),Y1:PRINT" ":LOCATE (X2-
1),Y2:PRINT"@":X1=X2:Y1=Y2
  170 IF ((X=X1)AND(Y=Y1)) THEN CLS:LOCATE 5,3:PRIN
T"I WIN":GOTO 190
  180 LOCATE (X-1),Y:PRINT"■":GOTO 60
  190 FOR I=1 TO 300:NEXT I:CLS:LOCATE 2,2:INPUT "P
LAY AGAIN(Y)";A$:IF A$="Y" THEN GOTO 10
  200 END
```

DEBUG

Whether or not you were pecked with the beak is determined by the IF statement in line 210. 100–140 decide whether or not a block is placed at a particular position.

Note that input of incorrect data destroys the screen display. Also, input of a number greater than 10 causes part of the screen display to remain. Delete the remaining portion using the SPACE key (if you like).